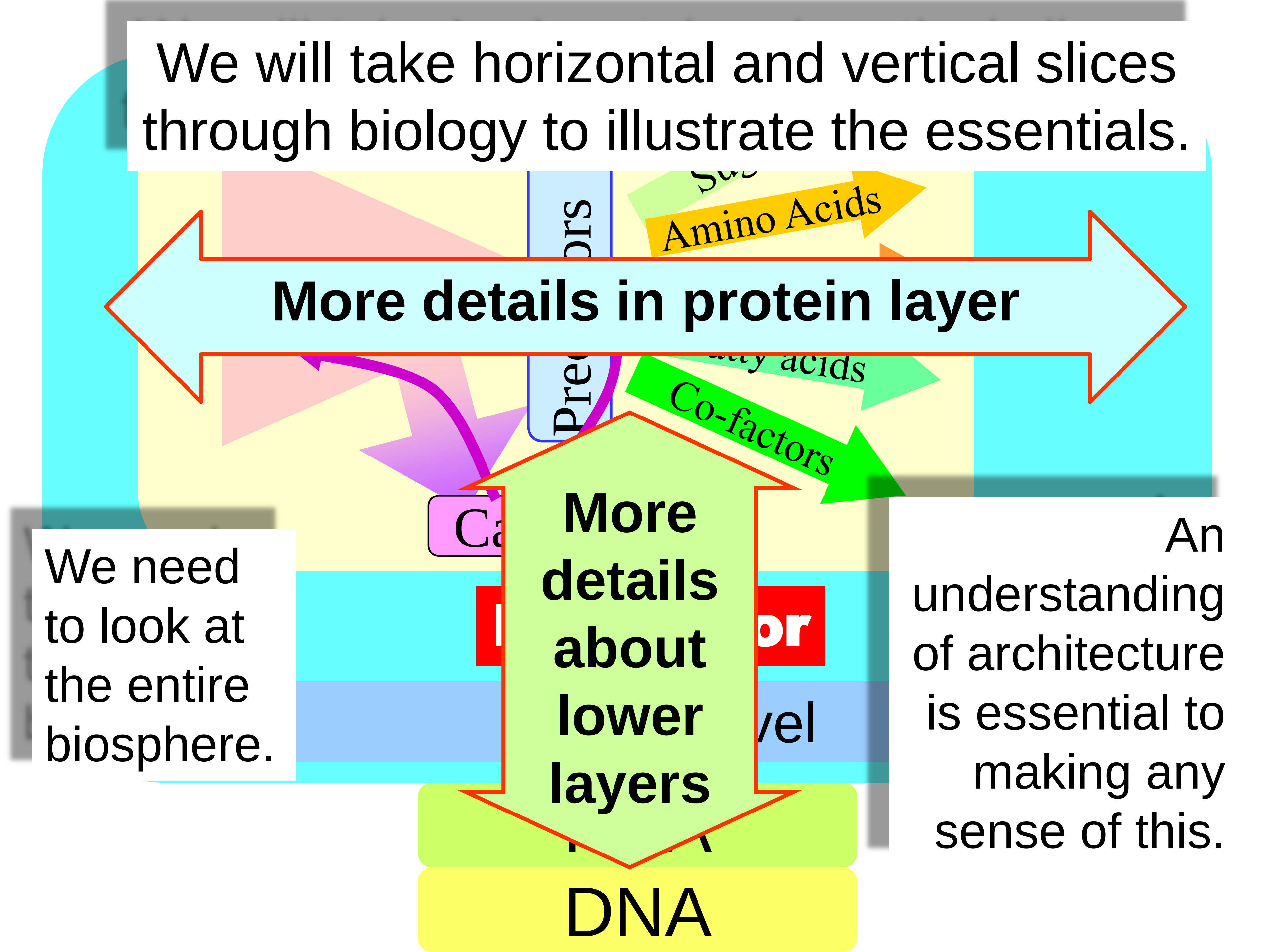
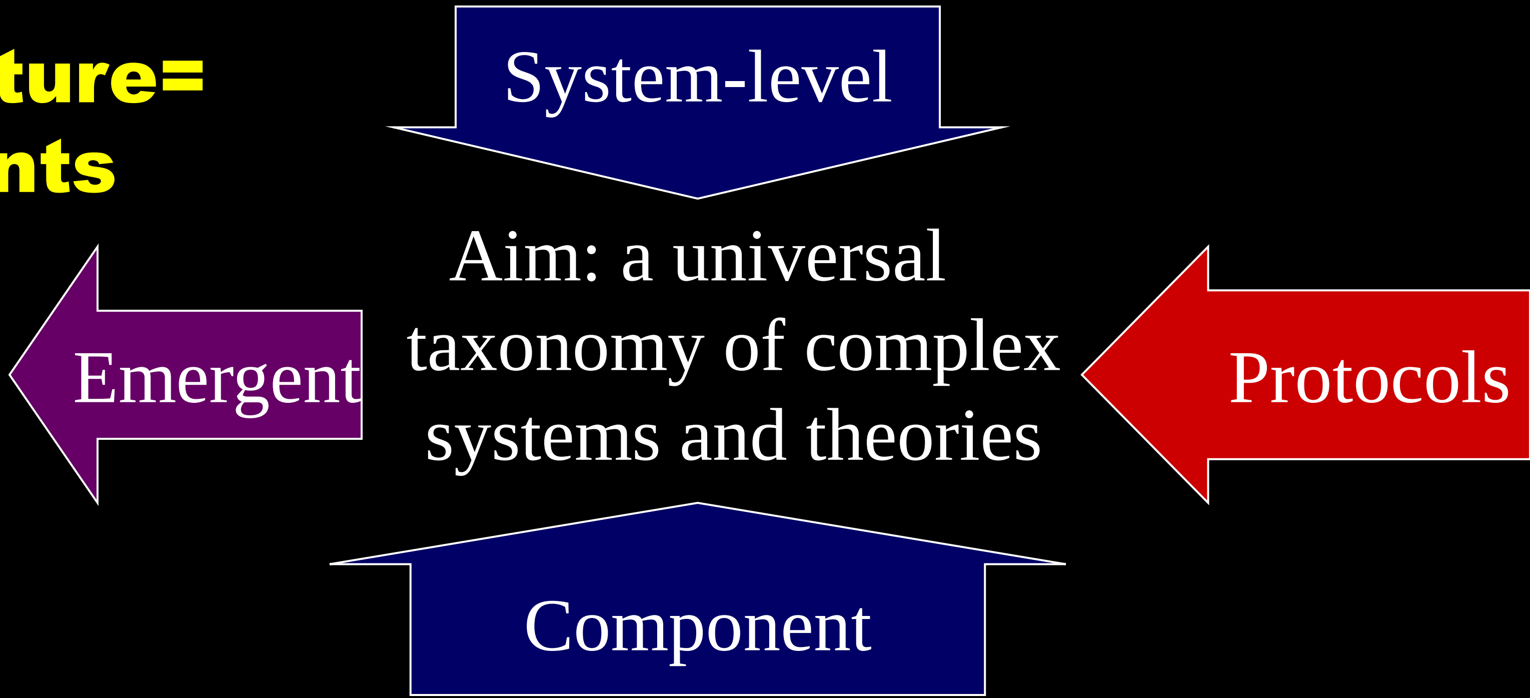
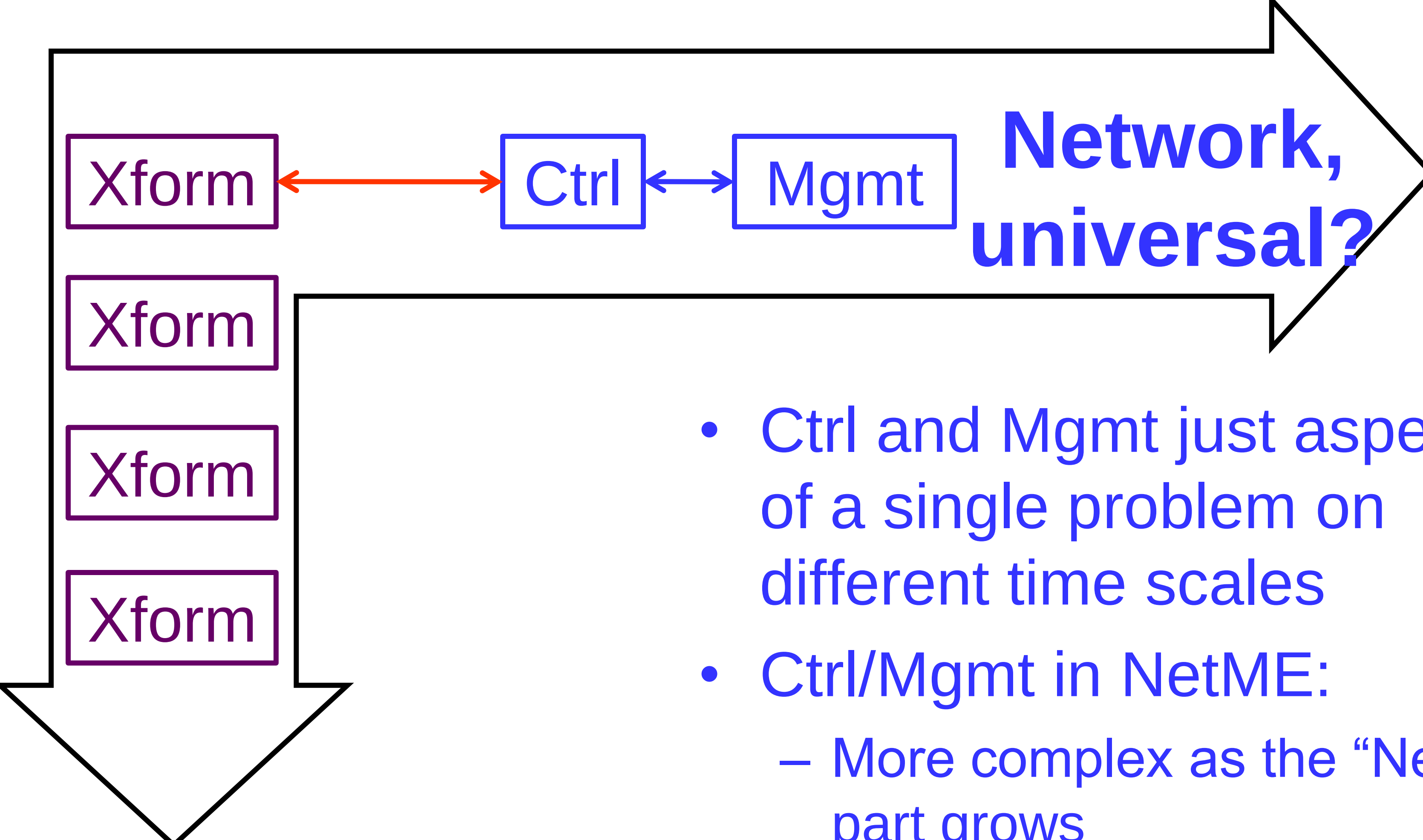

We will take horizontal and vertical slices through biology to illustrate the essentials.
Amino Acids
More details in protein layer
Co-factors
More details about lower layers
We need to look at the entire biosphere.
An understanding of architecture is essential to making any sense of this.
DNA

Architecture= Constraints

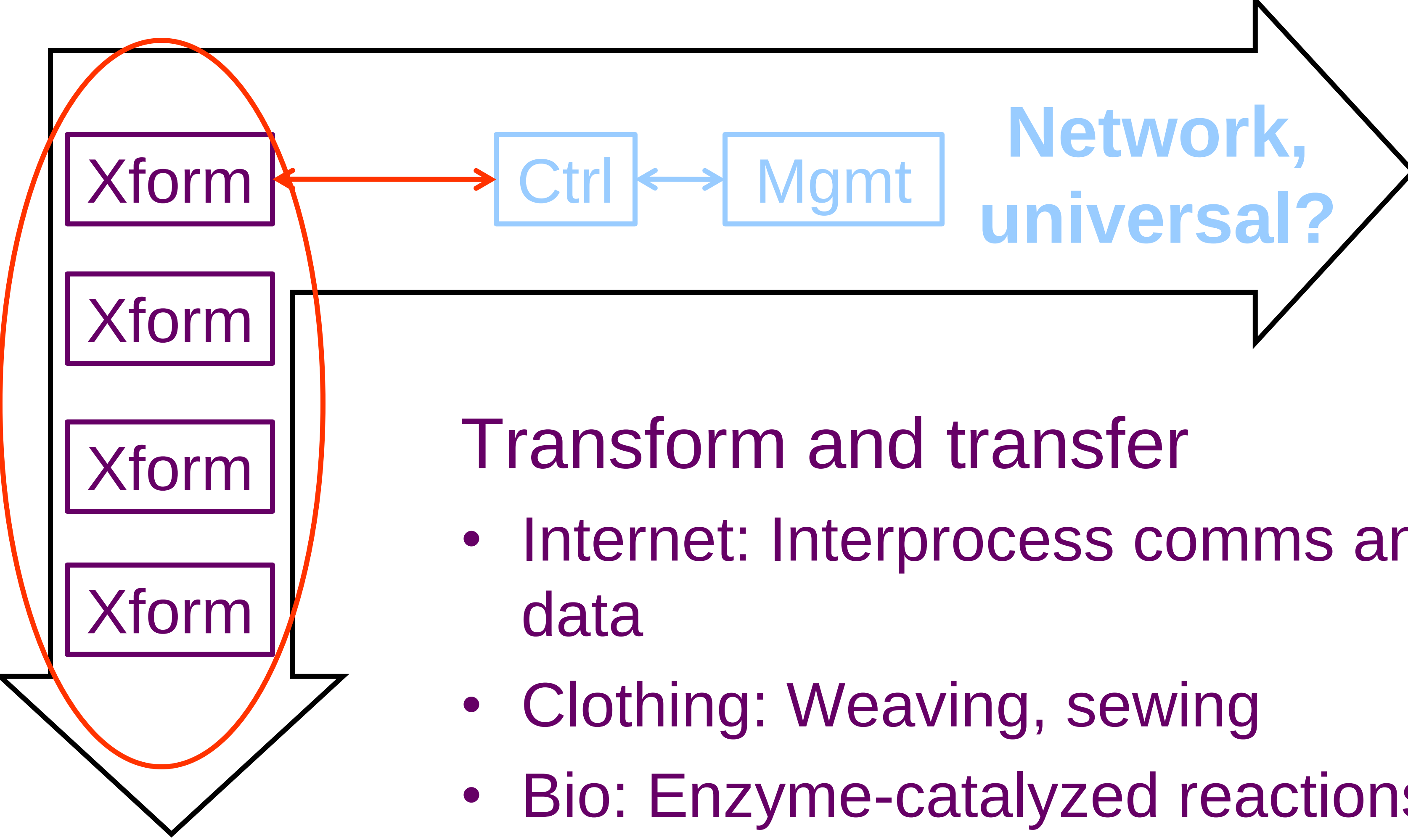


- Describe systems/components in terms of constraints on what is possible
- Decompose constraints into component, system-level, protocols, and emergent
- Not necessarily unique, but hopefully illuminating nonetheless



Domain specific, local

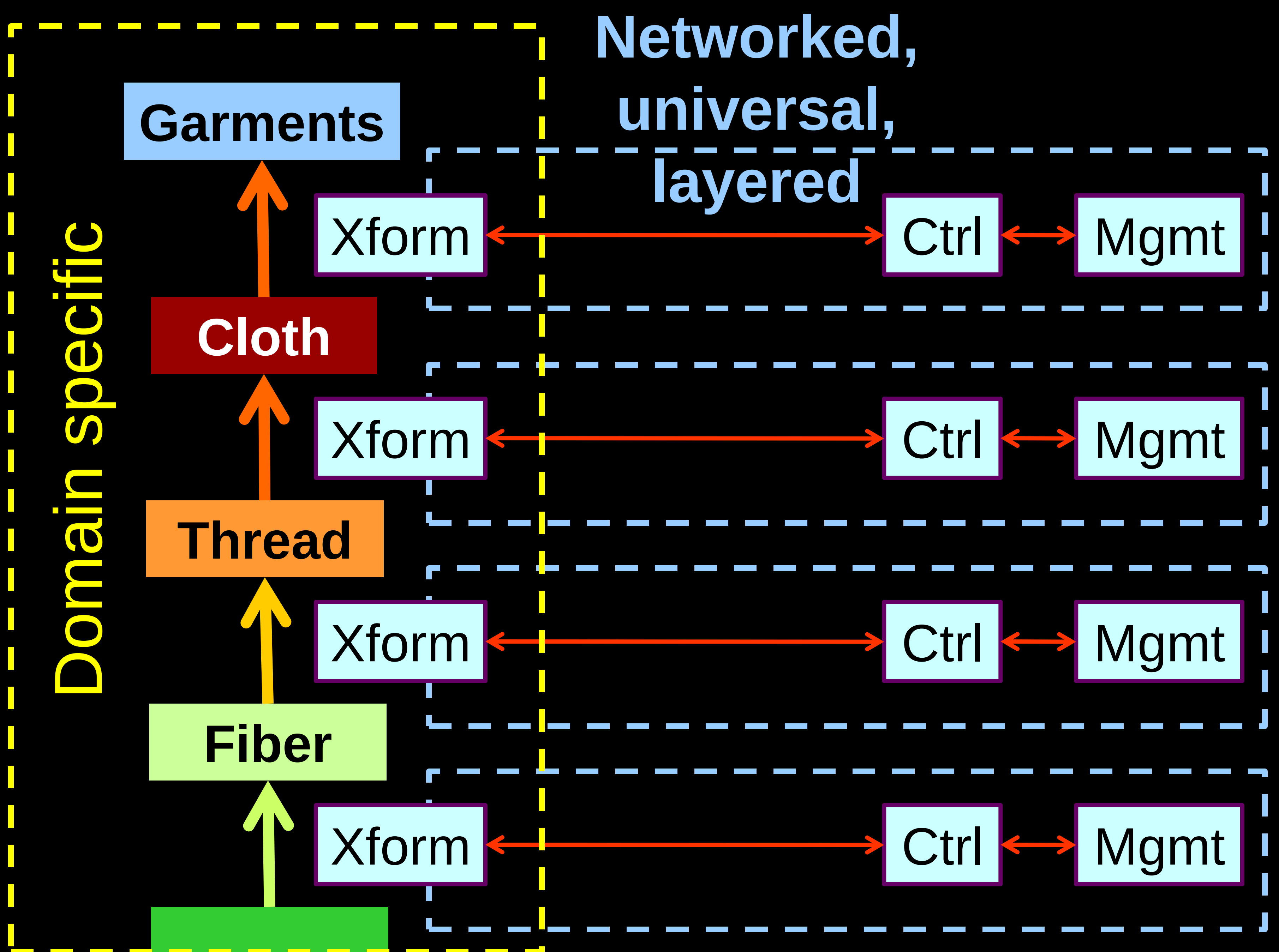
- Ctrl and Mgmt just aspects of a single problem on different time scales
- Ctrl/Mgmt in NetME:
  - More complex as the “Net” part grows
  - Will be our focus/goal of a unified theory
  - From physics to information to computation to control



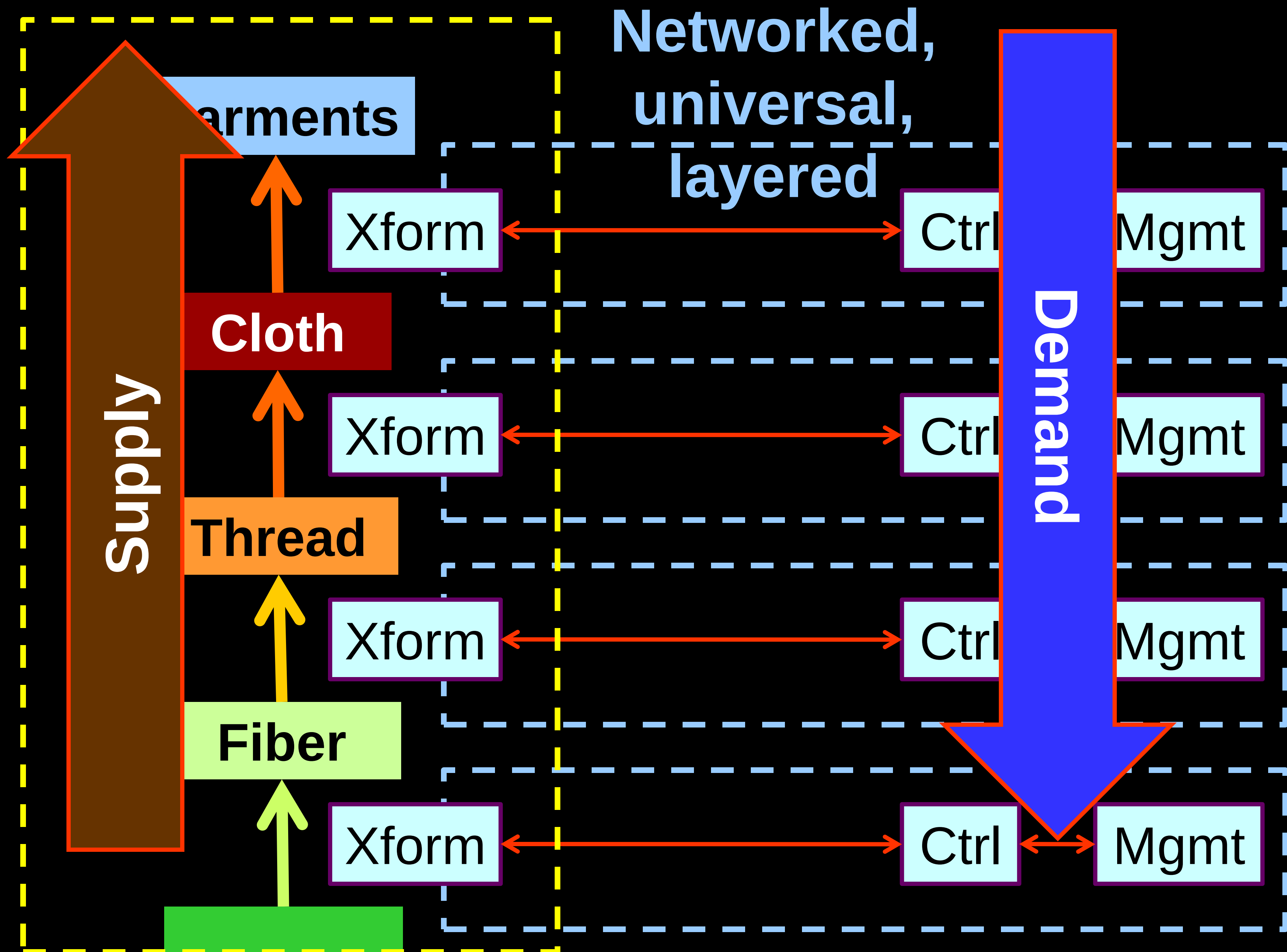
Domain specific, local

Transform and transfer

- Internet: Interprocess comms and data
- Clothing: Weaving, sewing
- Bio: Enzyme-catalyzed reactions and metabolites
- Lego: assembly and control



Networked, universal, layered
Domain specific
Garments
Cloth
Thread
Fiber
Xform
Ctrl
Mgmt
Xform
Ctrl
Mgmt
Xform
Ctrl
Mgmt
Xform
Ctrl
Mgmt

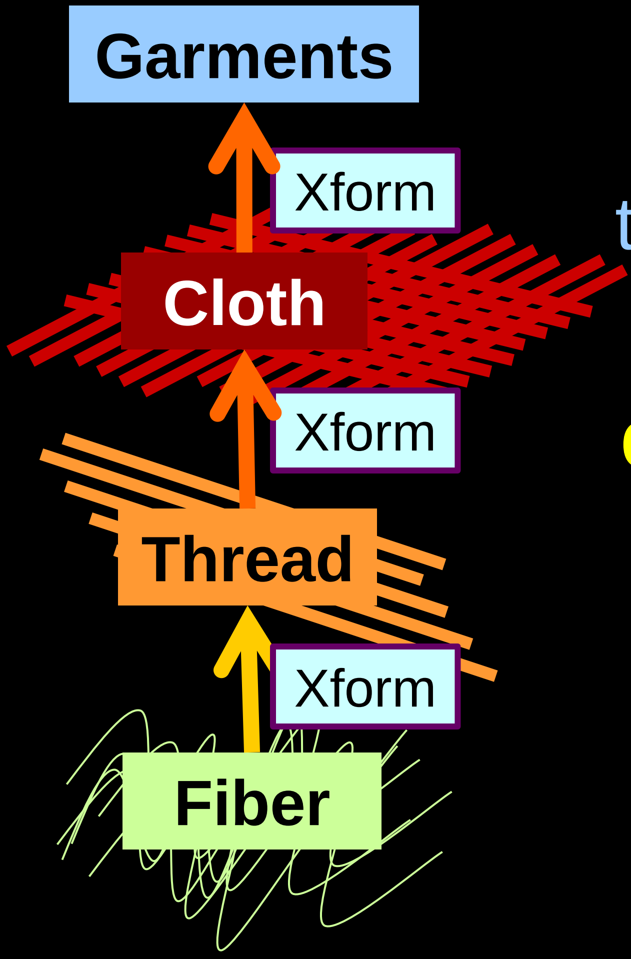


Networked, universal, layered
arments
Xform
Ctrl
Mgmt
Cloth
Xform
Ctrl
Mgmt
Demand
Supply
Thread
Xform
Ctrl
Mgmt
Fiber
Xform
Ctrl
Mgmt

Universal strategies?

Even though garments seem analog/continuous

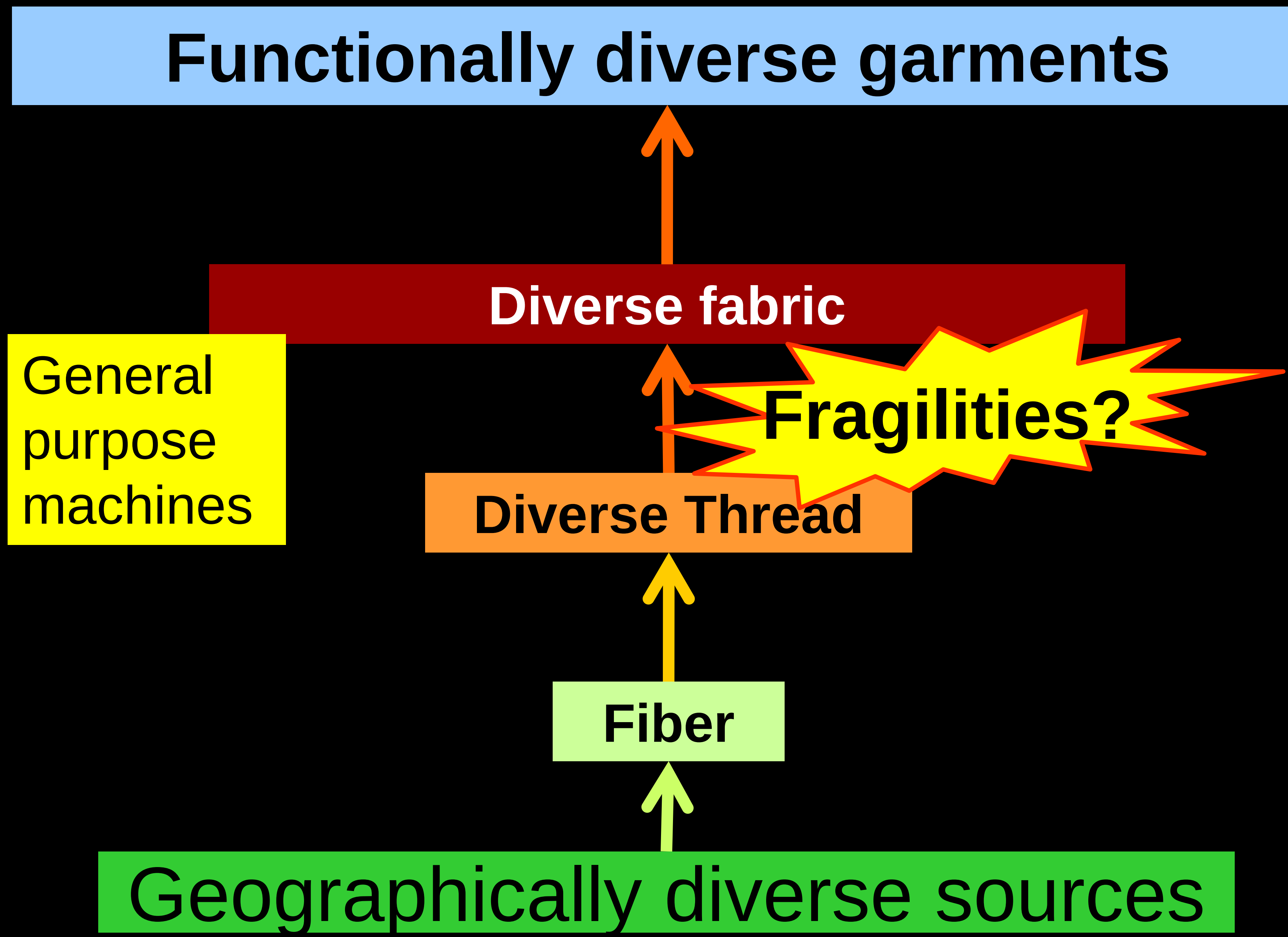
quantization for robustness



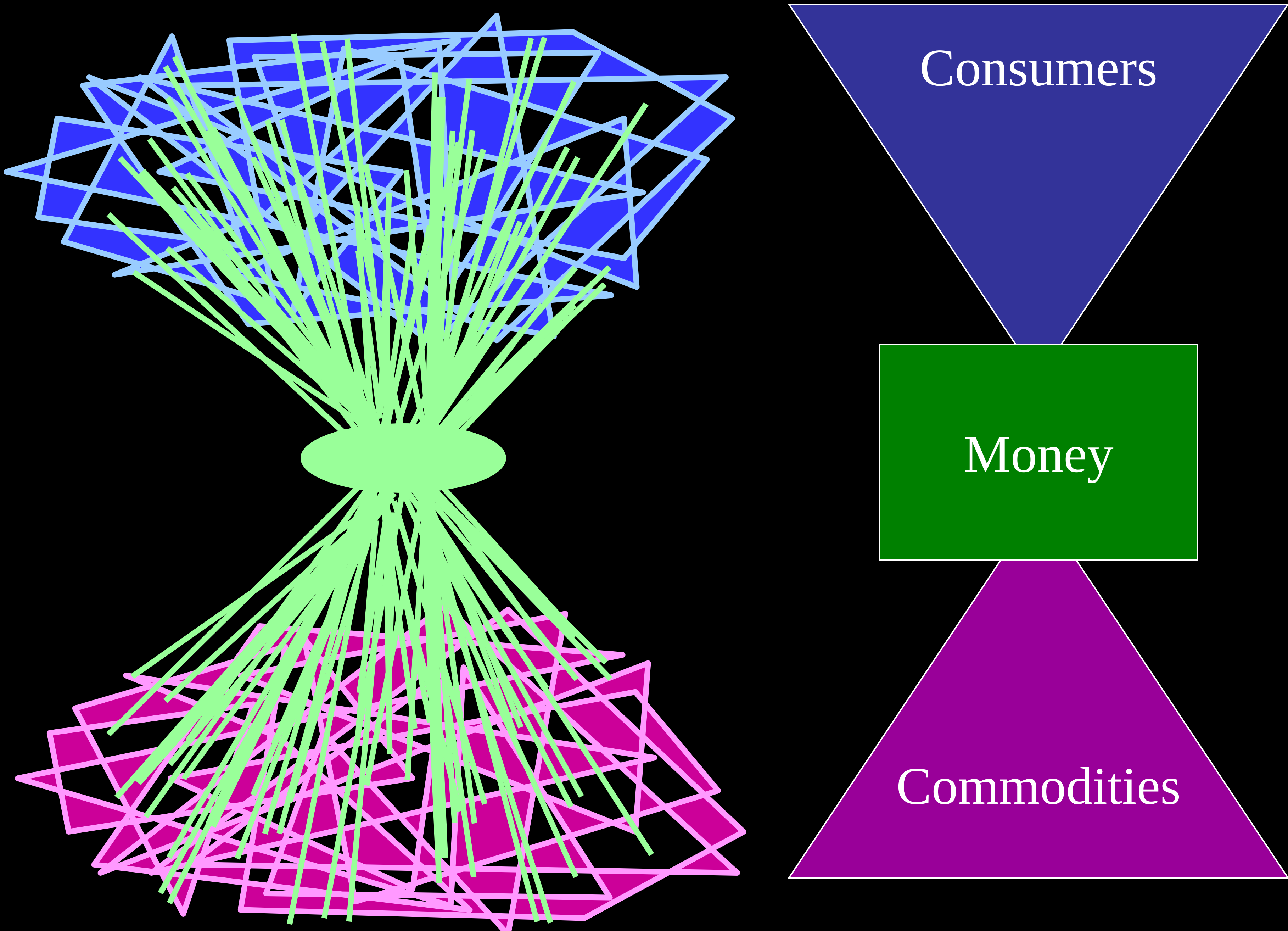
Garments have limited access to threads and fibers

constraints on cross-layer interactions

Prevents unraveling of lower layers

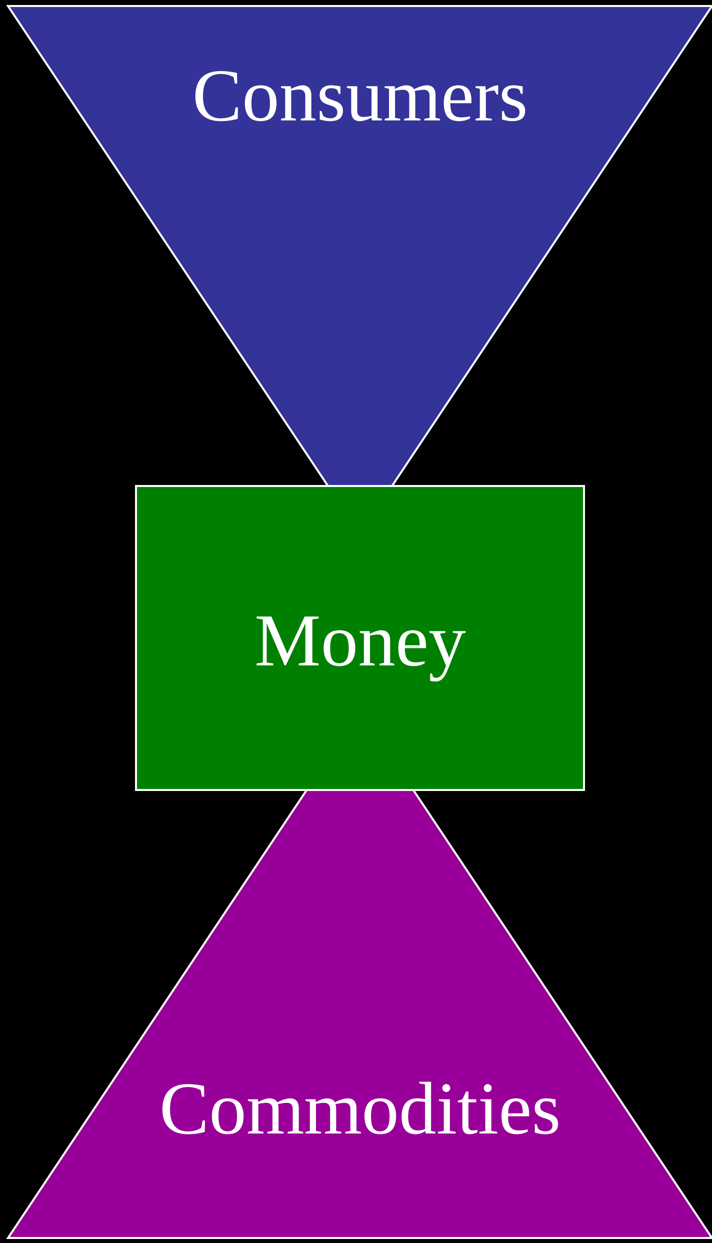

Functionally diverse garments
Diverse fabric
General purpose machines
Fragilities?
Diverse Thread
Fiber
Geographically diverse sources

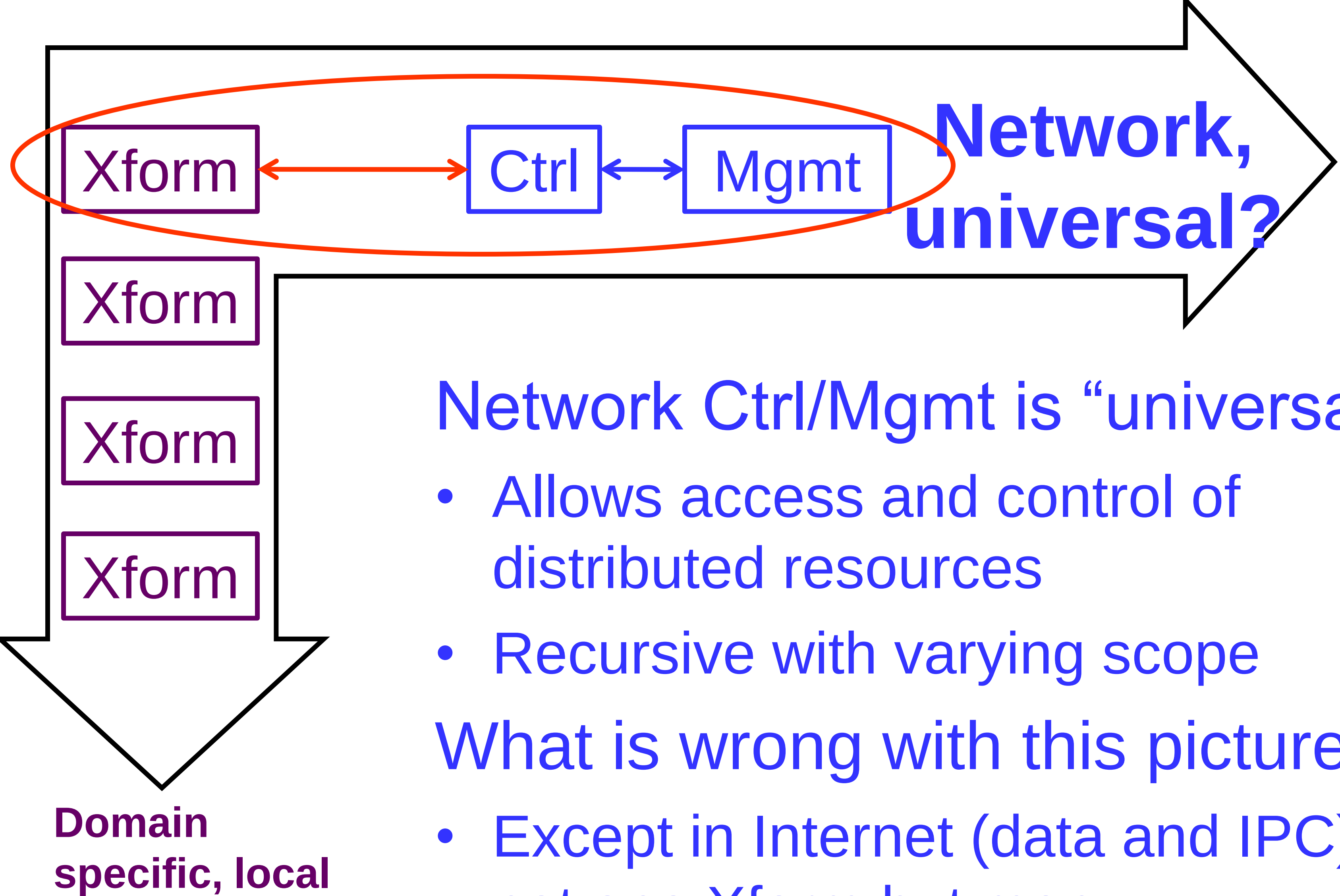
Money


Consumers
Money
Commodities

New fragilities

- Theft, counterfeiting, fraud, and “creative accounting” are now possible
- The beginning of a growing complexity-fragility spiral
- Complex legal infrastructure
- Law, banking, finance, Ponzi schemes, derivatives, credit default swaps, …





Network Ctrl/Mgmt is “universal”

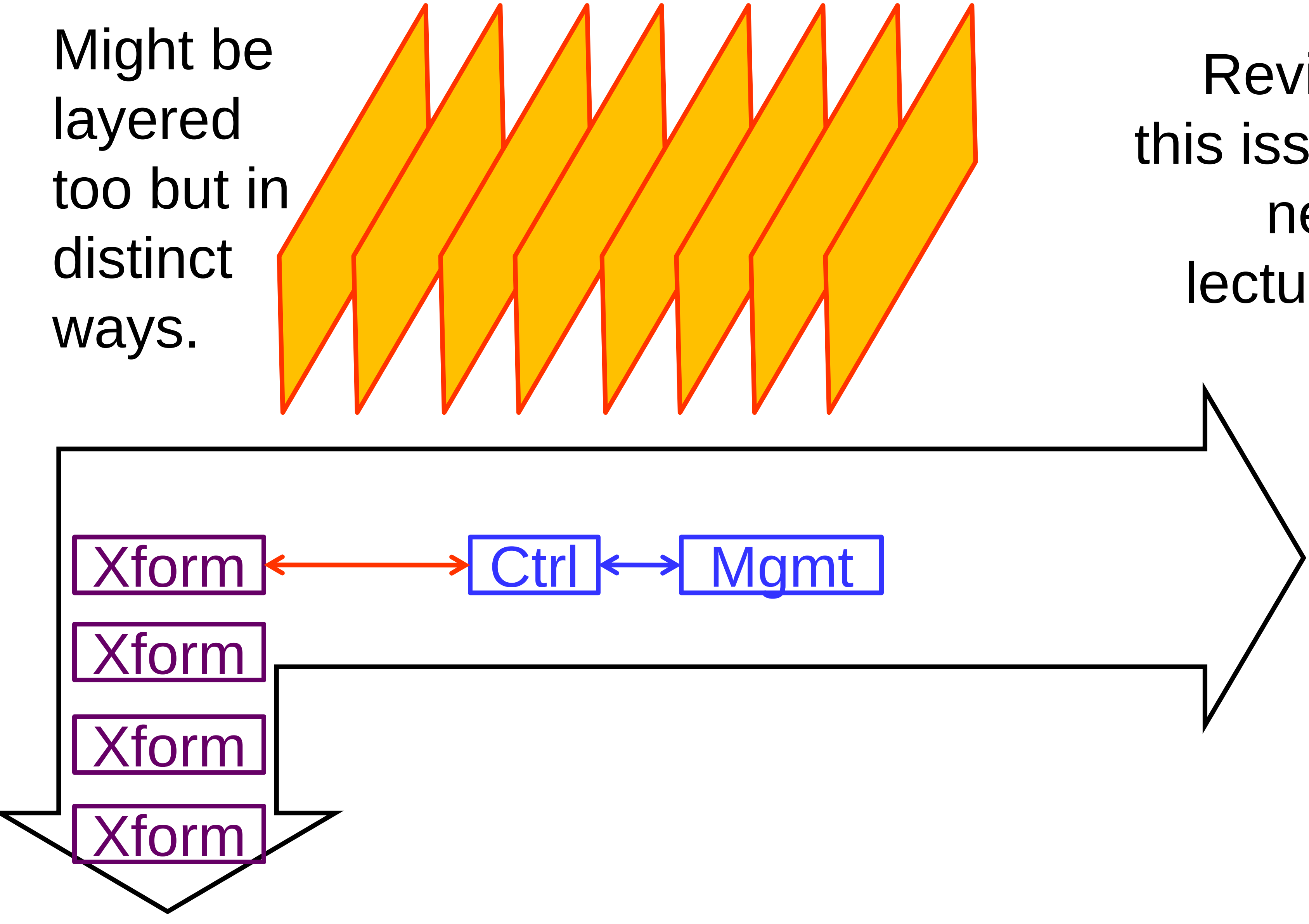
- Allows access and control of distributed resources
- Recursive with varying scope

What is wrong with this picture?

- Except in Internet (data and IPC), not one Xform but many
- Usually a platform for another architecture

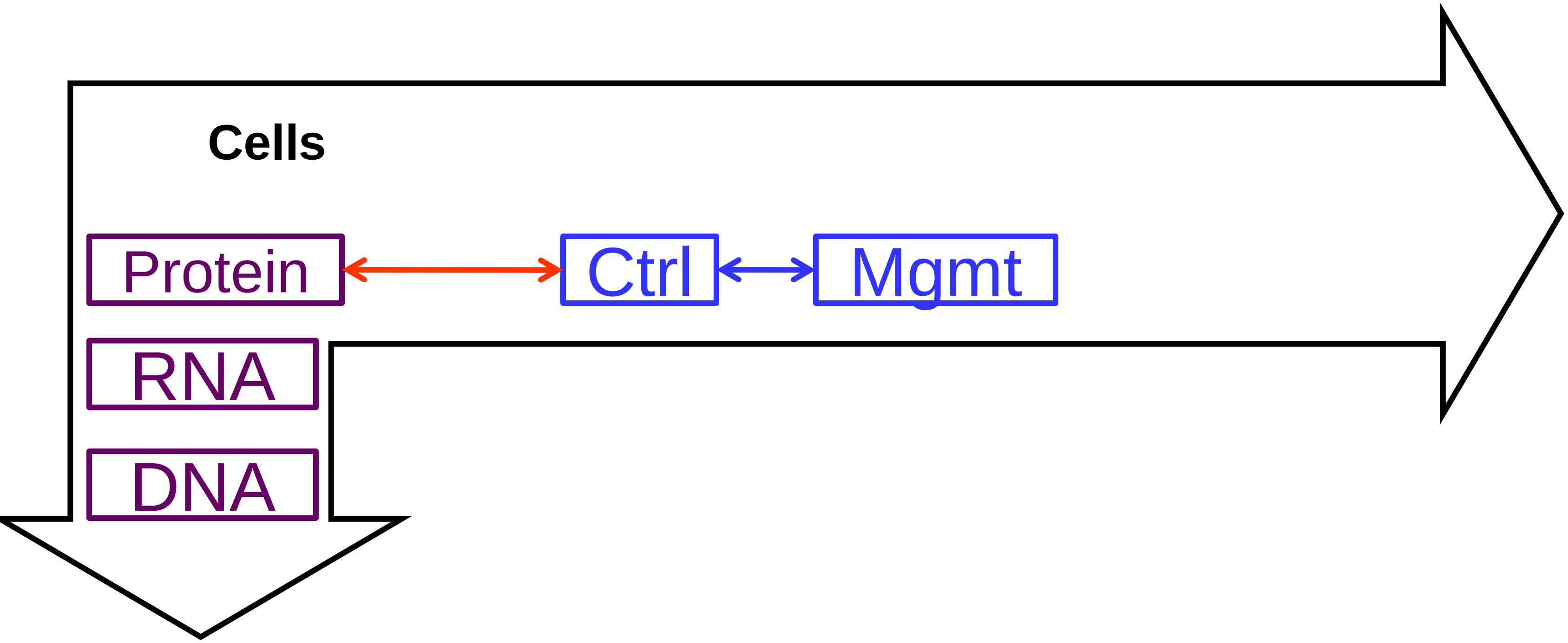
Usually a platform for other architectures

Might be layered too but in distinct ways.

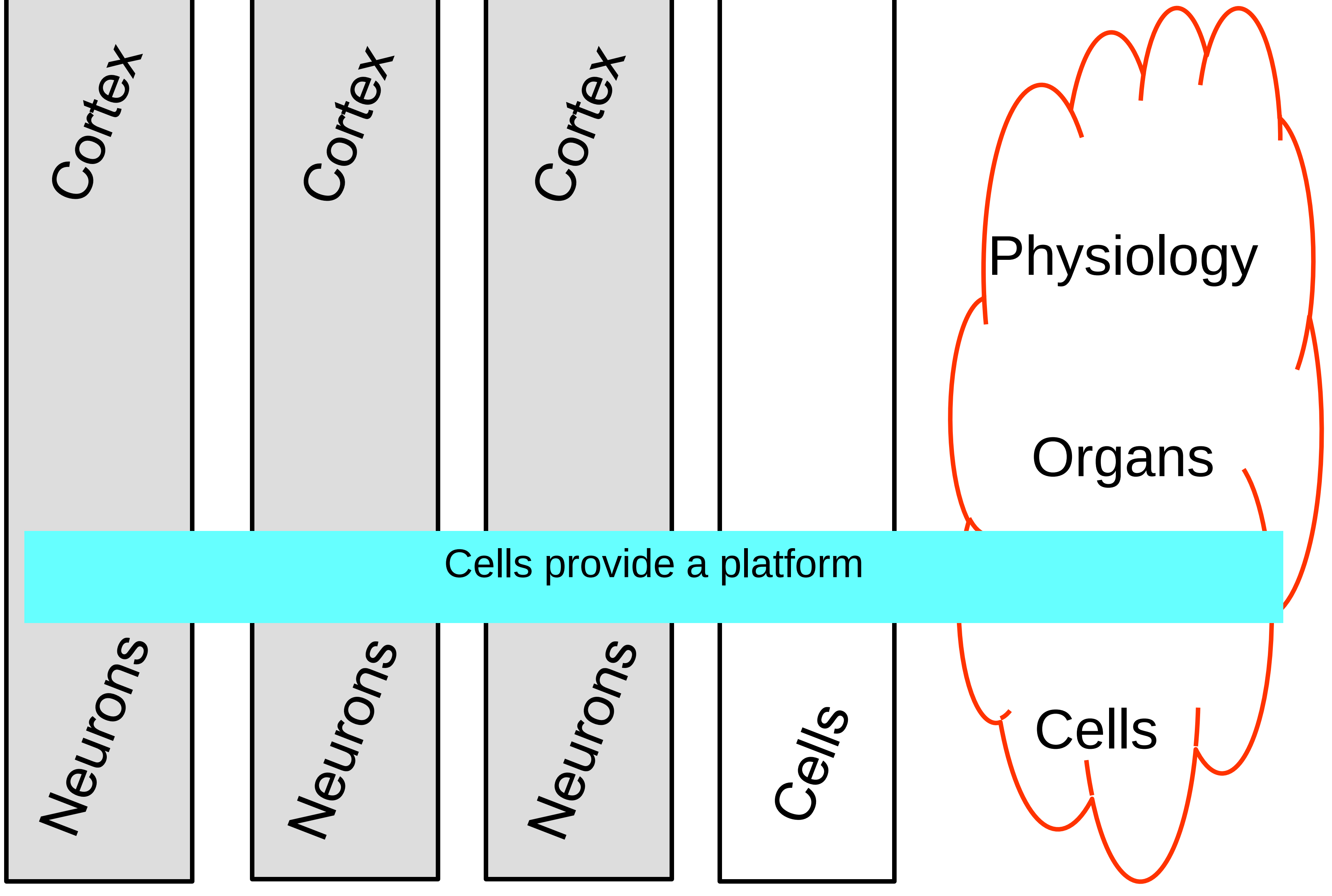


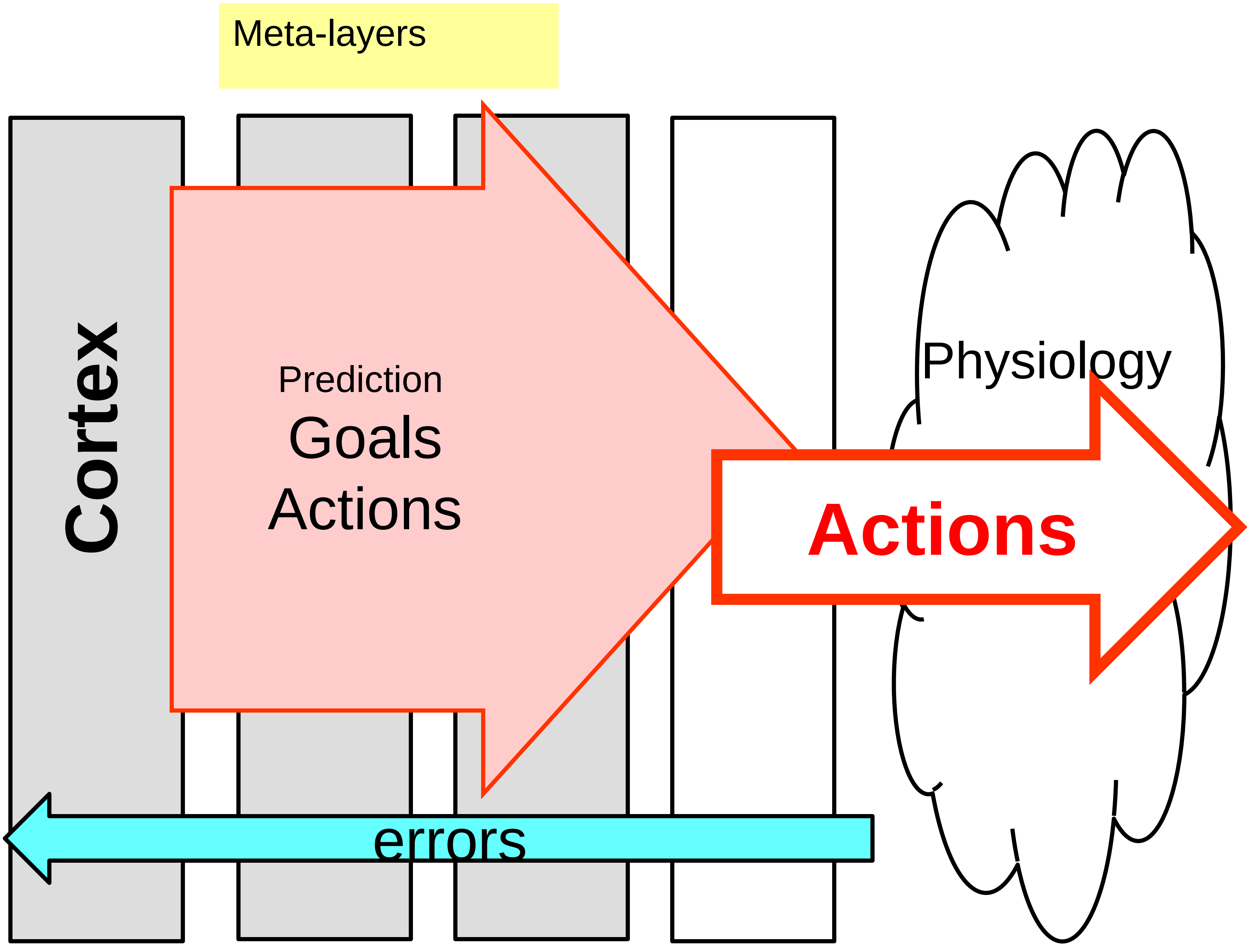
Revisit this issue next lecture.

For example, cells have internal layering that provides a platform, for a completely separate layering at the organ level

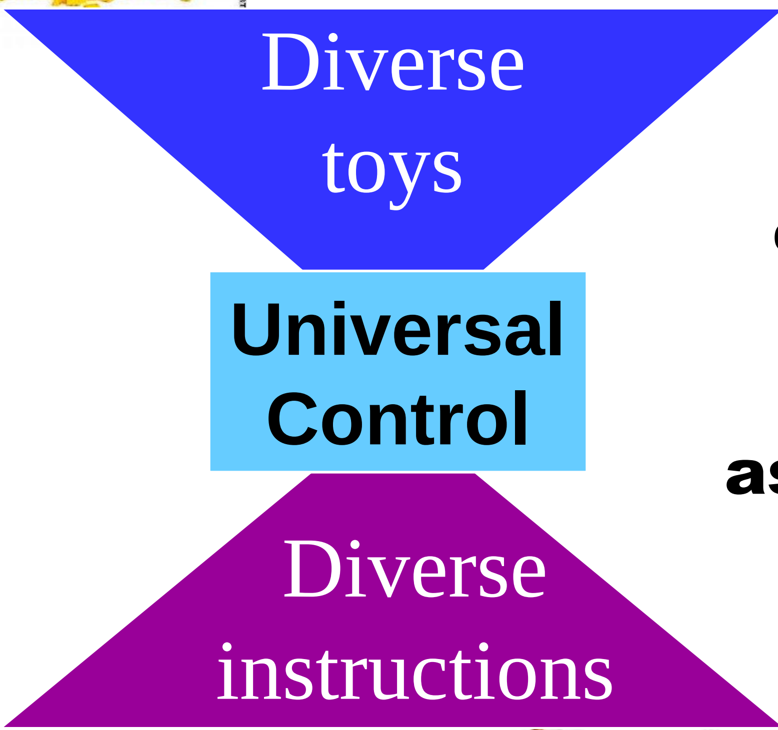


Layering of neural control



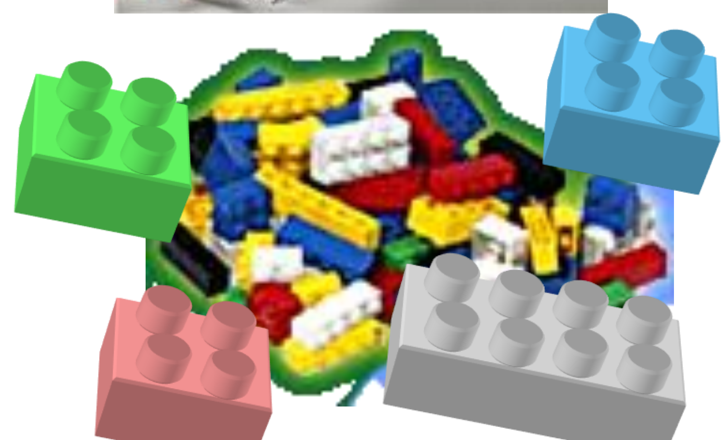

Meta-layers
Cortex
Prediction
Goals
Actions
Physiology
Actions
errors

Lego hourglass



control

assembly



Robust yet fragile

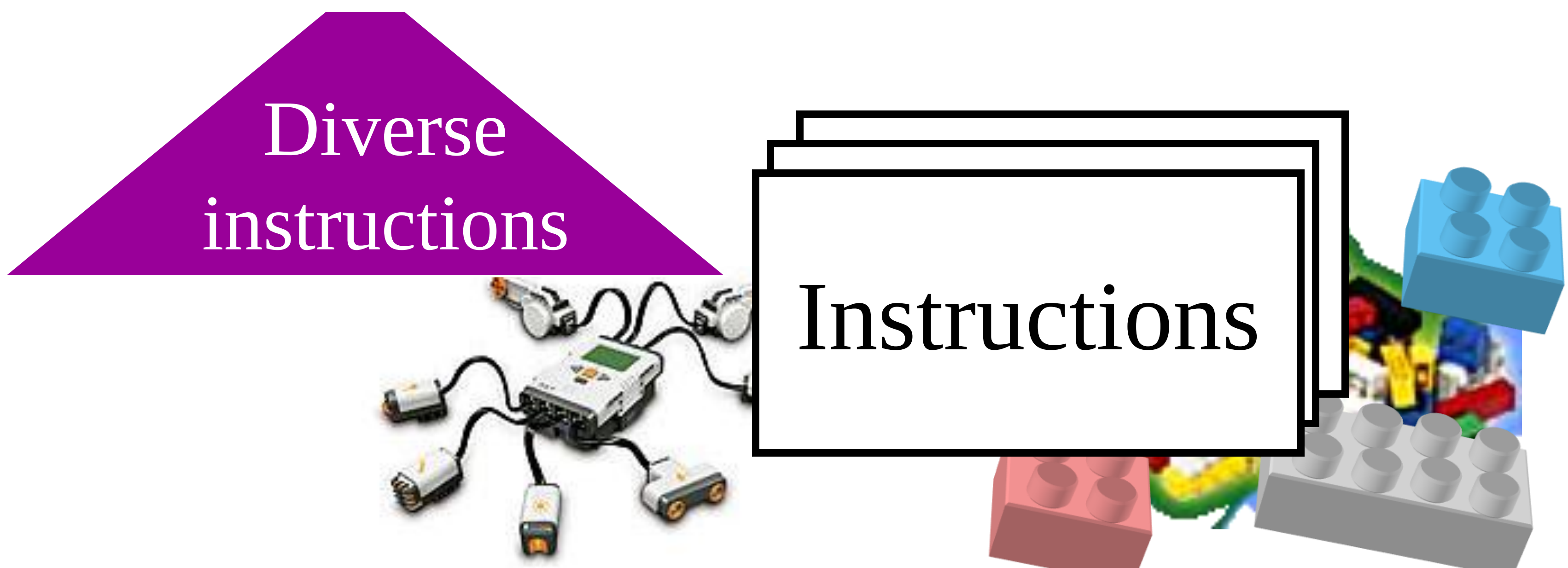
Extremes of

- Robust yet fragile
- Simplicity and complexity
- Constrained and flexible
- Frozen and evolvable
- Digital and analog
- Diverse and conserved

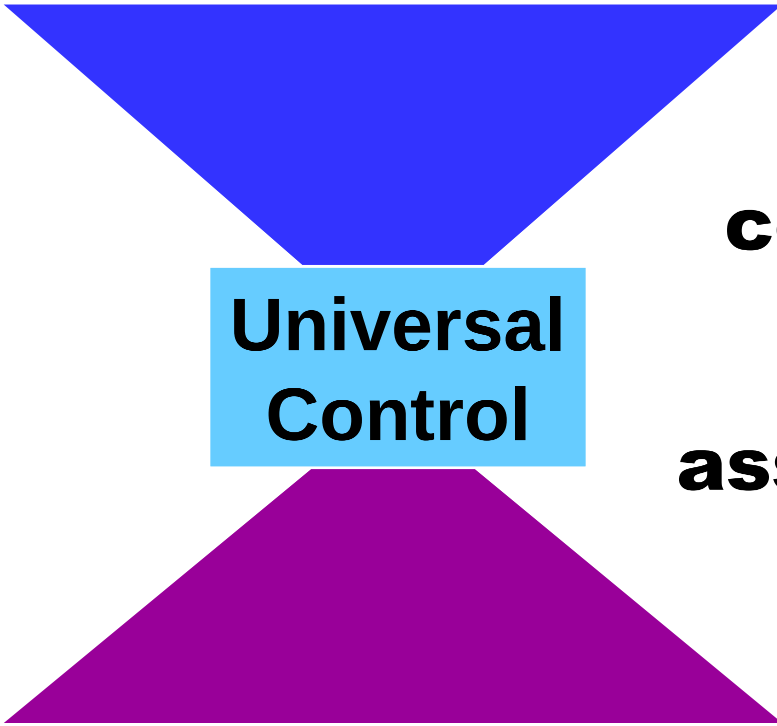
Lego hourglass

Diverse
toys

Lego hourglass



Lego hourglass

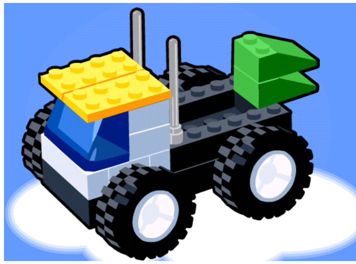


control

assembly

Lego hourglass

Huge variety



Standardized mechanisms
Highly conserved

control

assembly

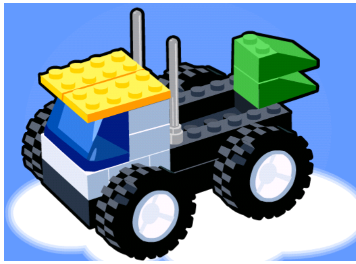
Huge variety

Instructions

Lego

Huge variety

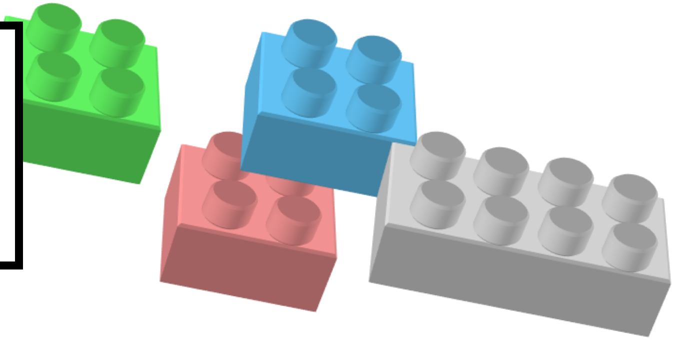
Limited environmental uncertainty needs minimal control

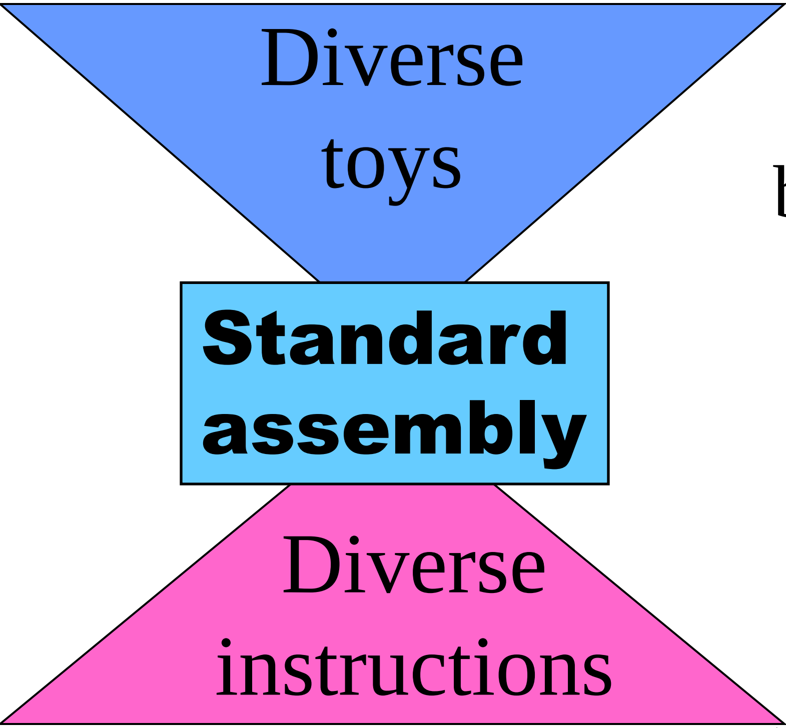


Standard assembly

Huge variety

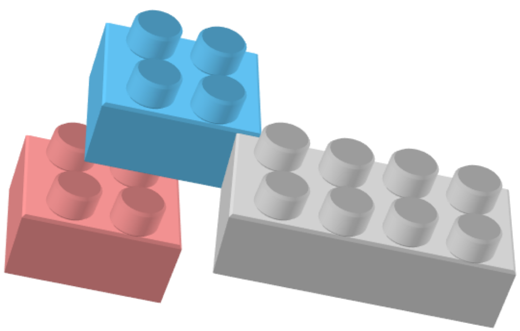
Instructions

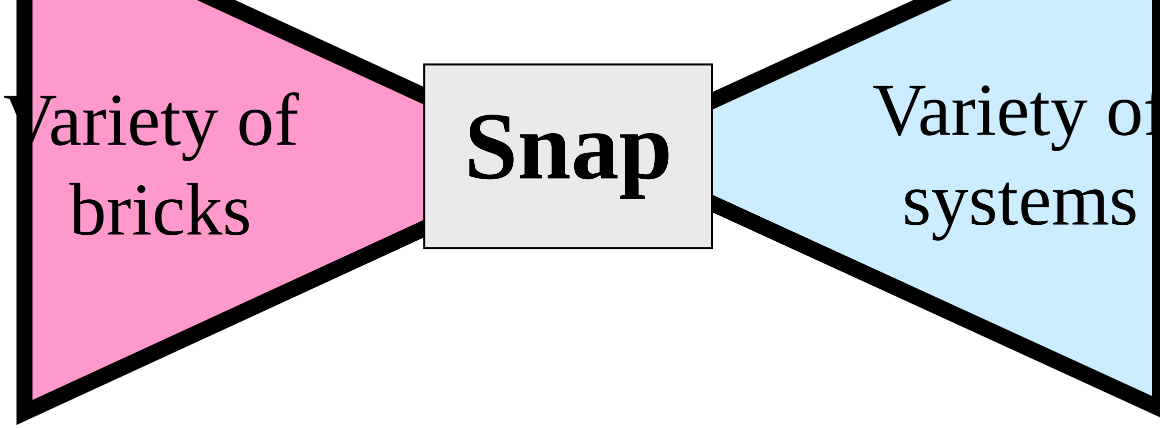




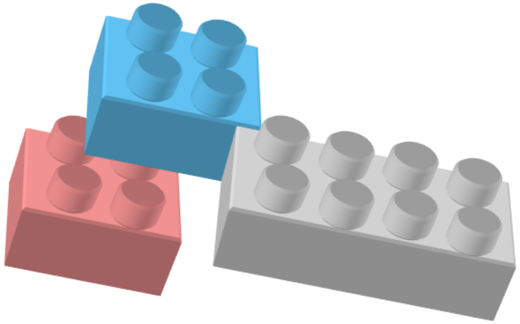
Question: what is the difference between hourglass and bowtie here?

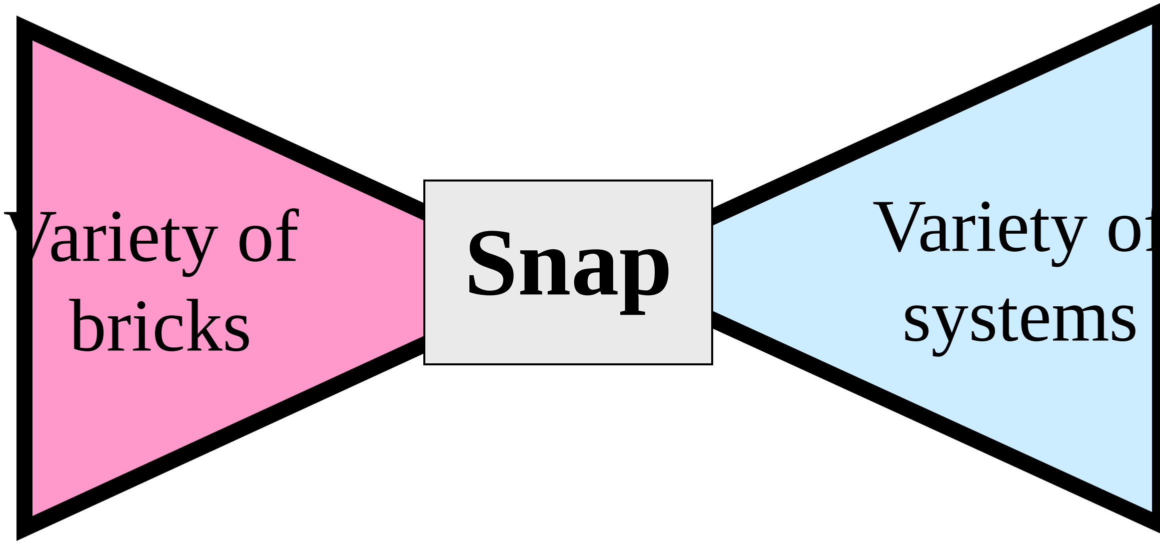
A “minimal” setting to address this issue.

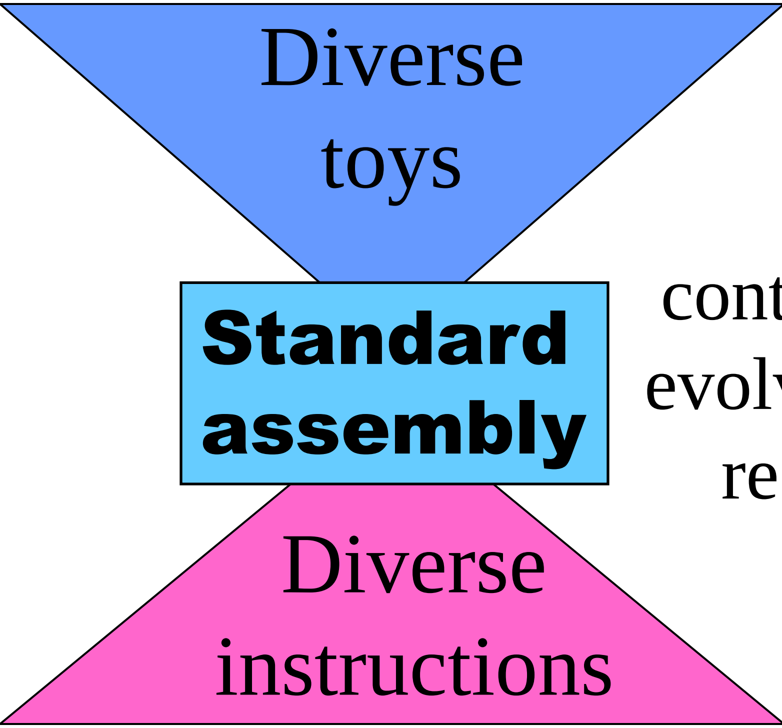




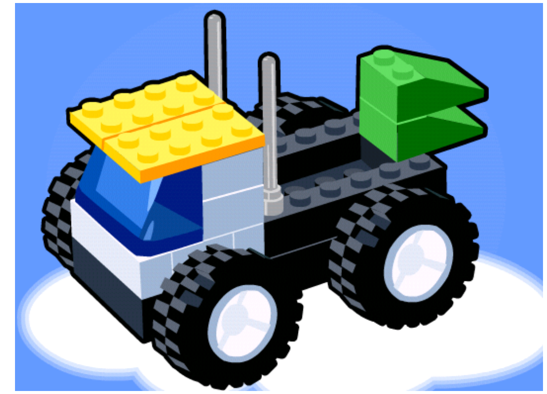
The snap is a static interface specification.



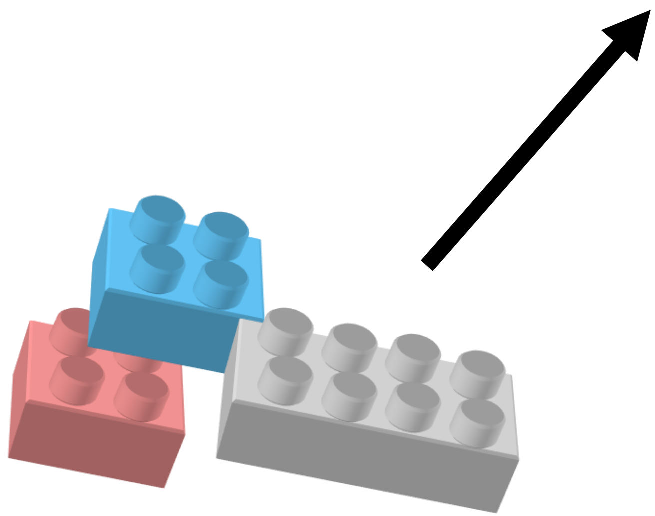


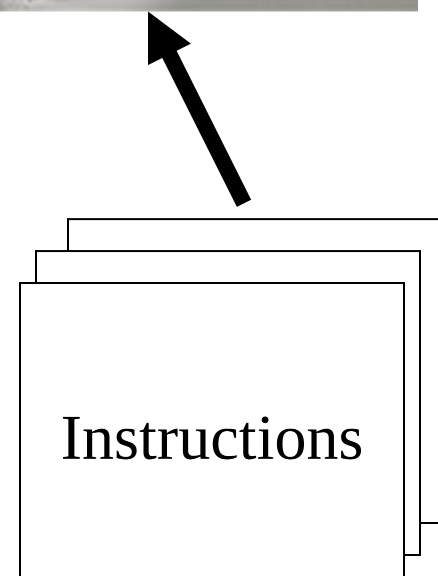


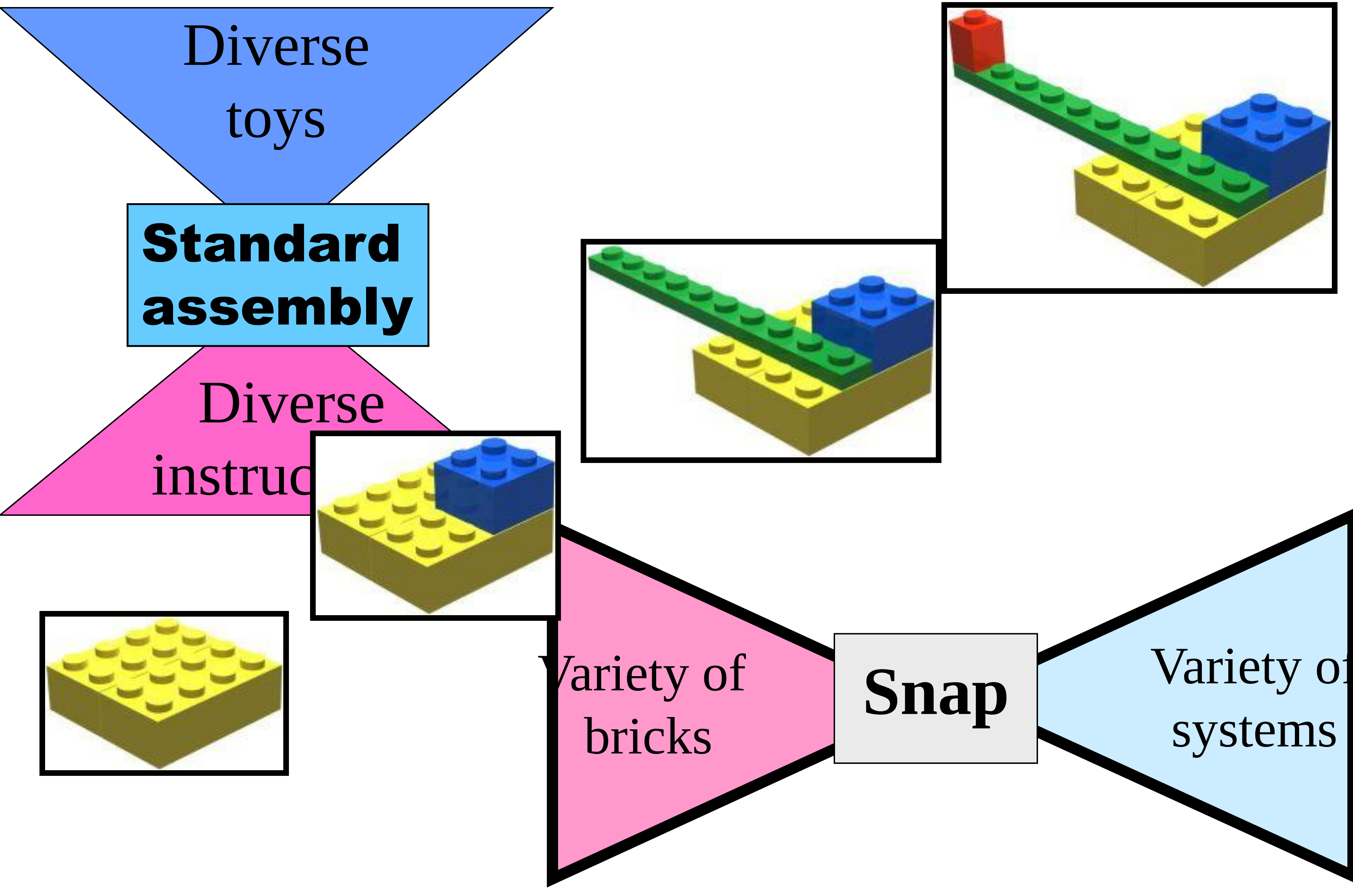
Assembly is a control process that evolves in time, and respects the snap, but adds to it.



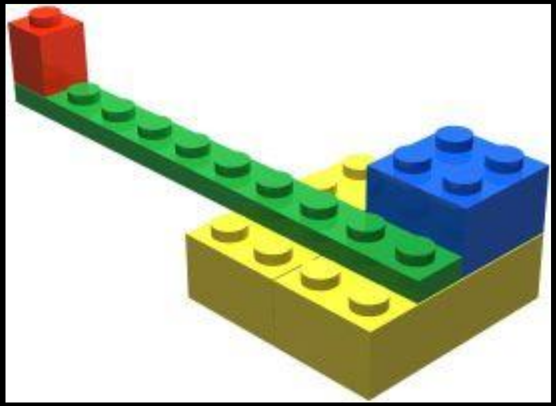
It inputs instructions and components and outputs assembled systems.

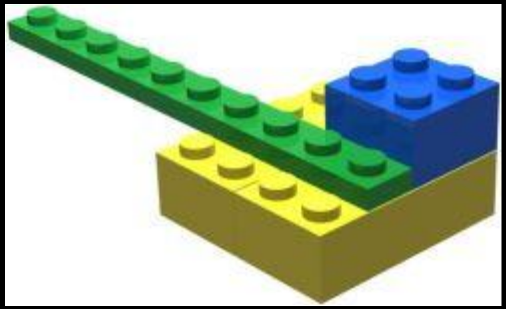


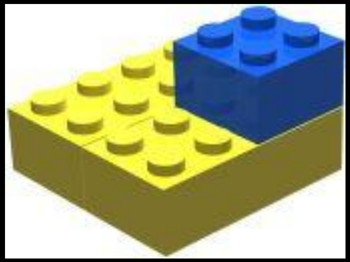


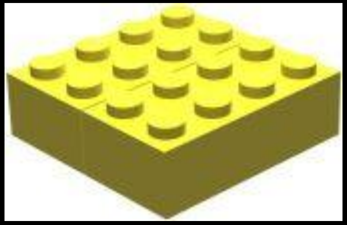

Diverse
toys
Standard
assembly
Diverse
instruc
ariety of
bricks
Snap
Variety o
systems

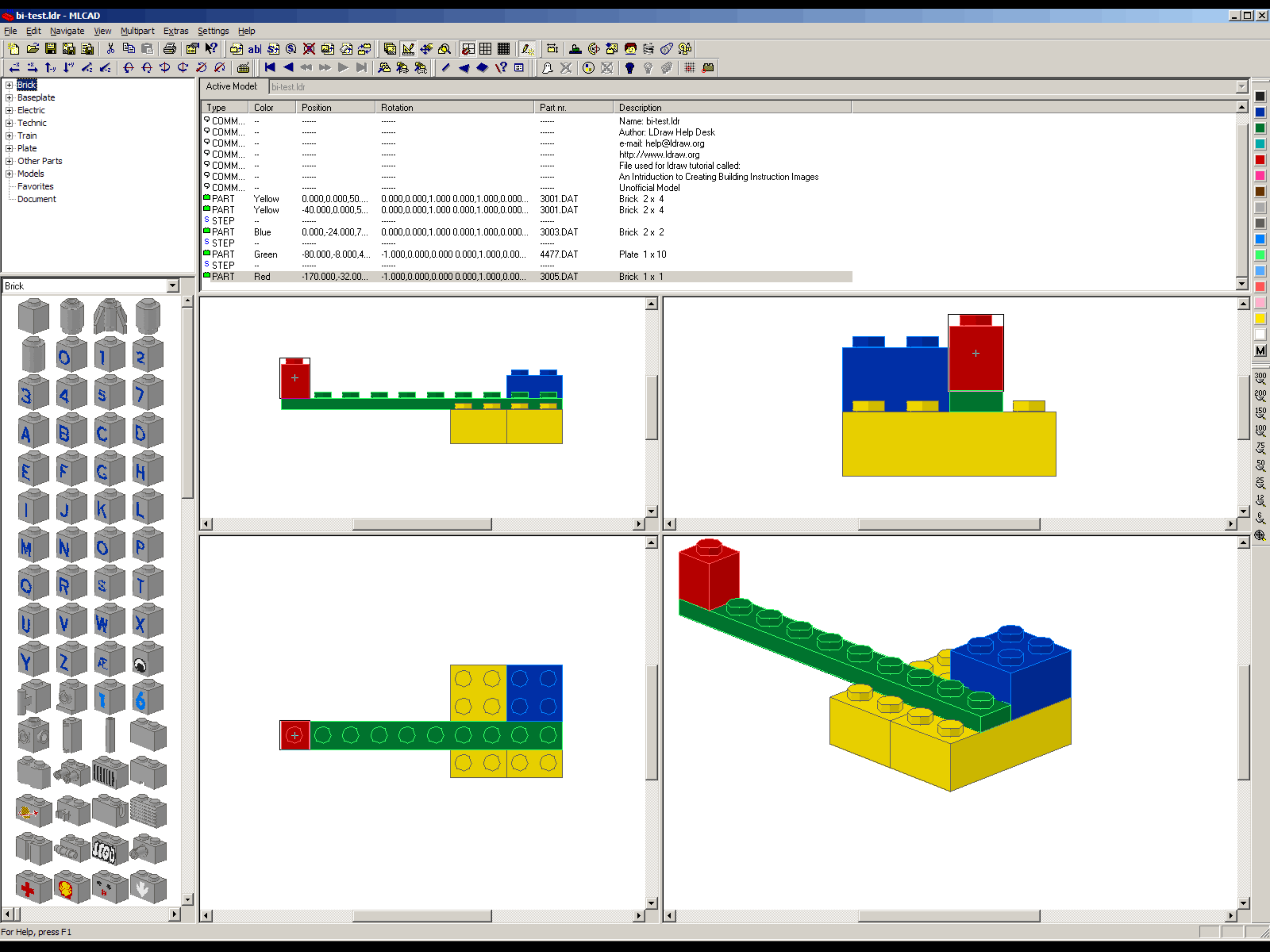
Computer aided design tools

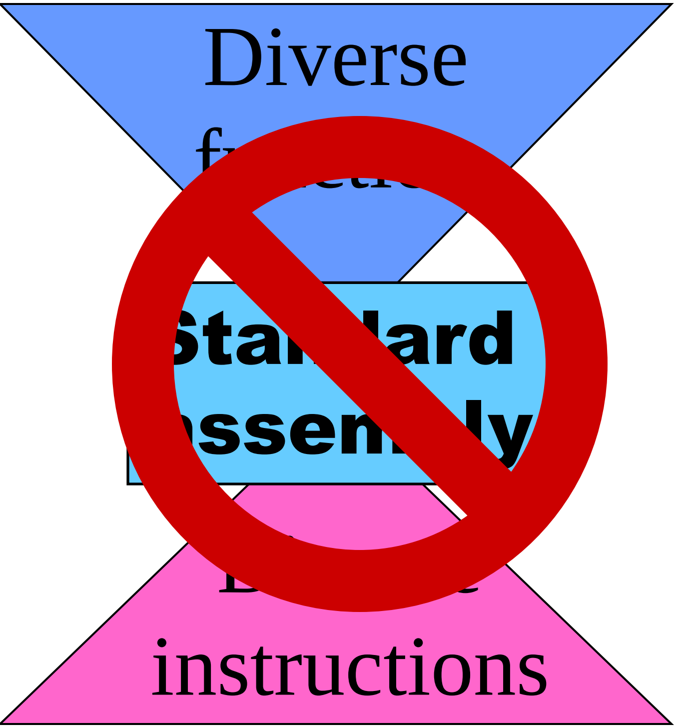




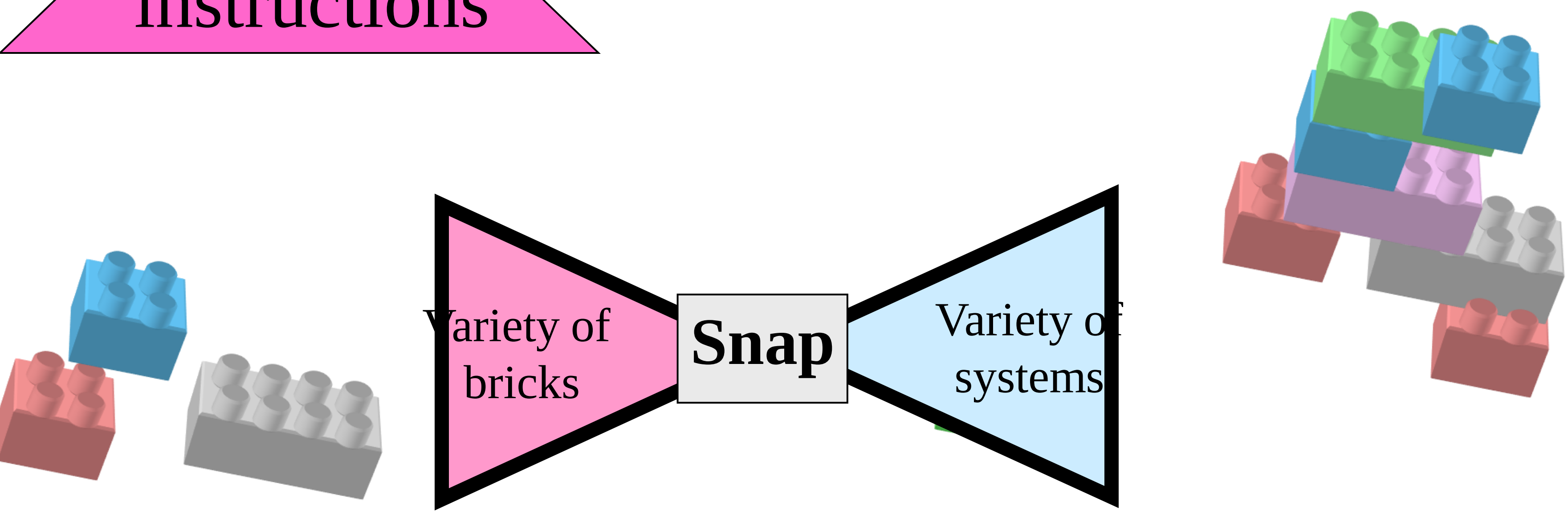


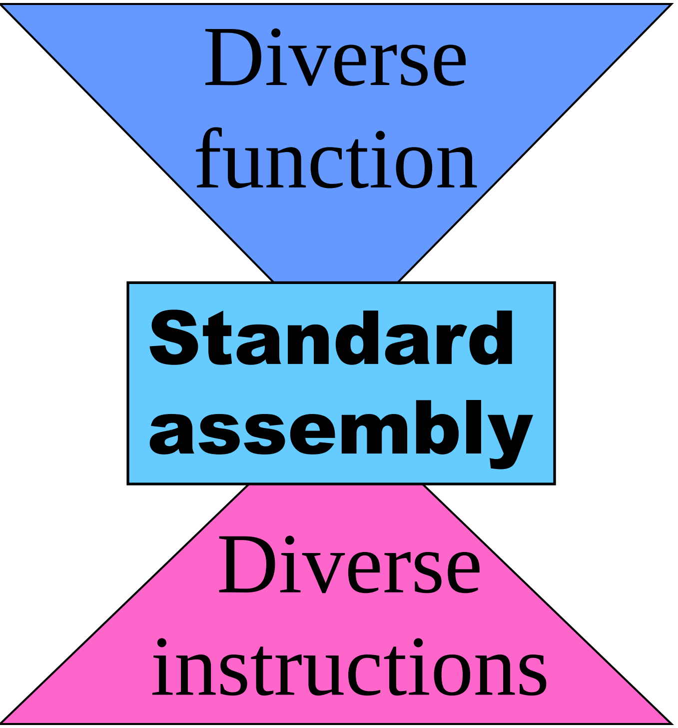



bi-test.ldr - MLCAD
File Edit Navigate View Multipart Extras Settings Help
Brick
Baseplate
Electric
Technic
Train
Plate
Other Parts
Models
Favorites
Document
Brick
Active Model:
bi-test.ldr
Type
Color
Position
Rotation
Part nr.
Description
COMM... -- Name: bi-test.ldr
COMM... -- Author: LDraw Help Desk
COMM... -- e-mail: help@ldraw.org
COMM... -- http://www.ldraw.org
COMM... -- File used for ldraw tutorial called:
COMM... -- An Intriduction to Creating Building Instruction Images
COMM... -- Unofficial Model
PART Yellow 0.000,0.000,50.... 0.000,0.000,1.000 0.000,1.000,0.000... 3001.DAT Brick 2 x 4
PART Yellow -40.000,0.000,5... 0.000,0.000,1.000 0.000,1.000,0.000... 3001.DAT Brick 2 x 4
STEP --
PART Blue 0.000,-24.000,7... 0.000,0.000,1.000 0.000,1.000,0.000... 3003.DAT Brick 2 x 2
STEP --
PART Green -80.000,-8.000,4... -1.000,0.000,0.000 0.000,1.000,0.00... 4477.DAT Plate 1 x 10
STEP --
PART Red -170.000,-32.00... -1.000,0.000,0.000 0.000,1.000,0.00... 3005.DAT Brick 1 x 1
For Help, press F1

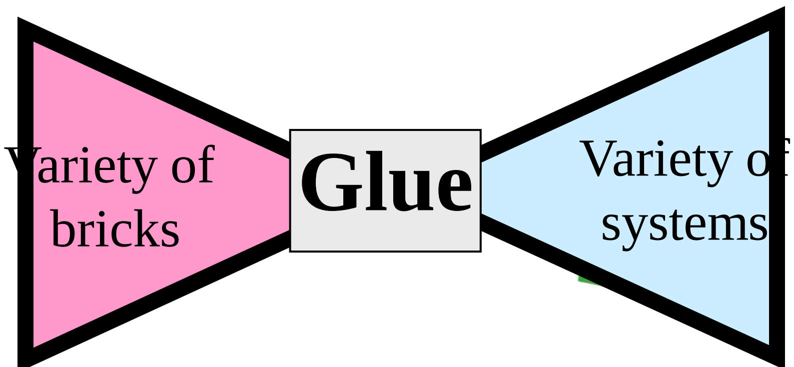


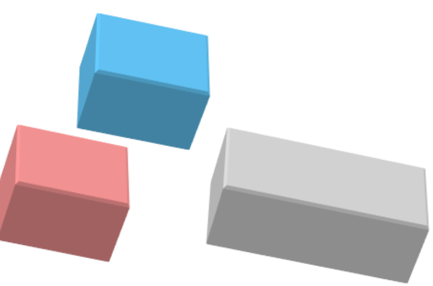
Random, uncontrolled, snap connection of Lego parts yields “nonfunctional” toys.

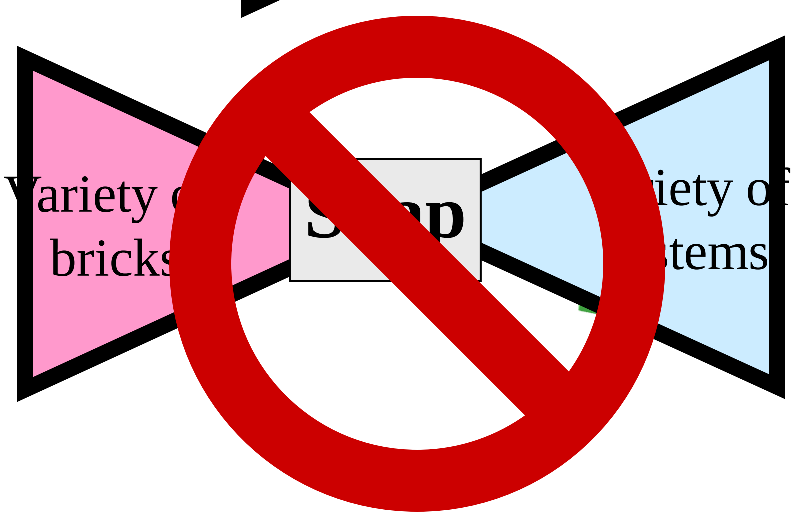


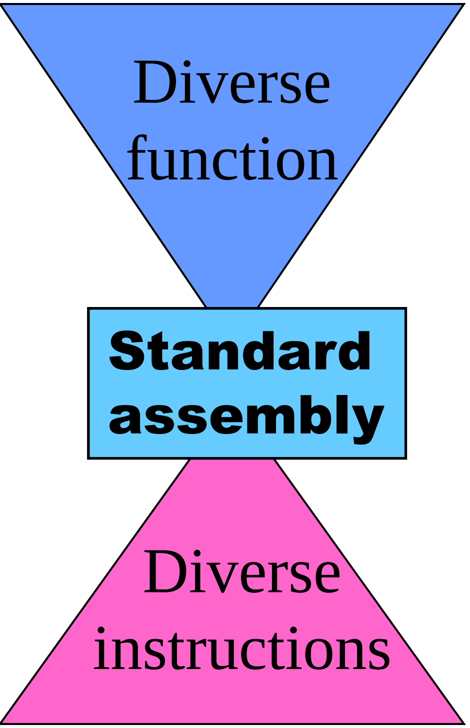


Loss of reuse, gain in robustness.



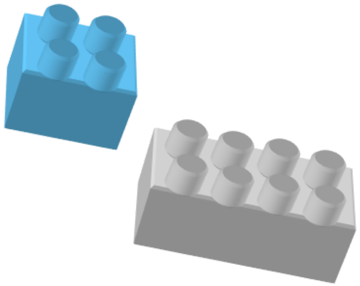


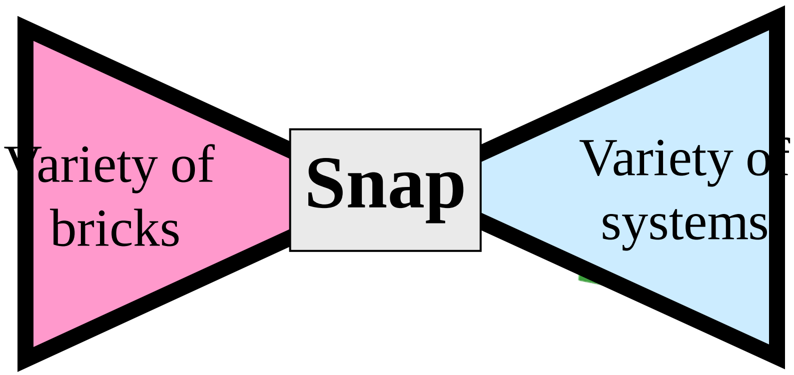


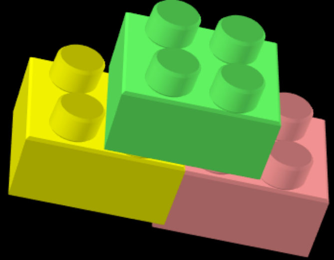


Robustness/ Evolvability

- A huge variety of new and different toys can be built
- From a huge variety of different components
- Both toys and components can be rearranged and added in new ways
- Yet fragile?





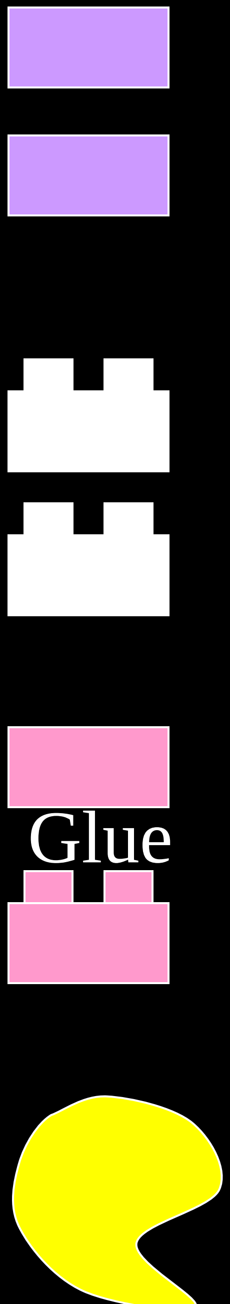


Yet fragile

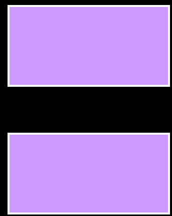
- Add or remove a tiny, indistinguishable amount of material from either side of a key interface.
- → ***Complete failure.***
- Other parts of the bricks may be nicked or cut with minimal impact
- This robust, yet fragile (RYF) feature of protocols is a candidate for a universal law
- "Layering" hides the snap in an assembly
- What robustness/fragility properties do alternative protocols have?

Lego ***system*** requirements

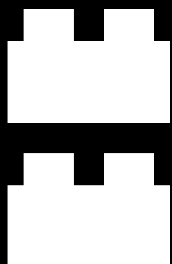
	Alternative designs?			
Performance				
Trauma				
Allowed connections				
Reuse				
Evolvable parts				
Evolvable systems				
Labor cost				
Parts cost				



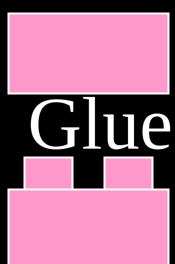
Consider some alternative interfaces and their tradeoffs…



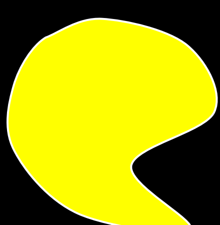
No interface. Simple blocks.



Standard interface. (Wild type.)

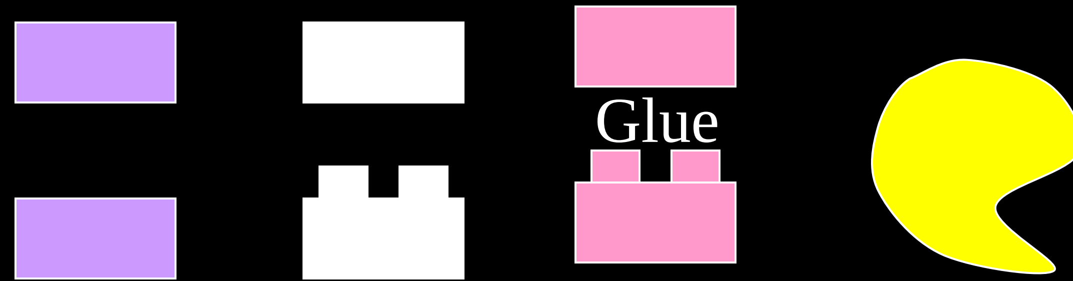


Add glue to hold the parts together.



Injection mold the whole toy from scratch.

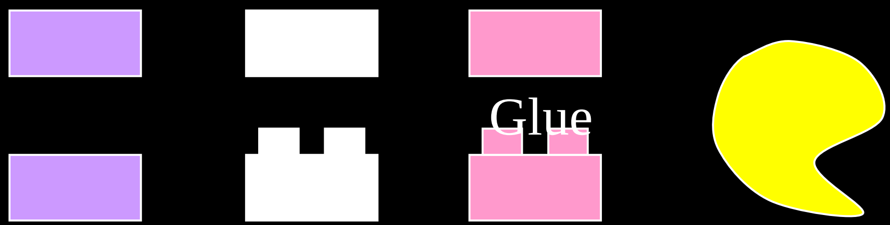
↑ better
↓ worse



	Smooth	WT	Glue	Mold
Performance	↓	---	---	↑↑
Trauma	↓↓	---	↑	↑↑
Allowed connections	↑↑	---	---	↓↓
Reuse	---	---	↓↓	↓↓
Evolvable parts	↓	---	---	↓
Evolvable systems	↓↓	---	---	↓
Labor cost	↓	---	↓	↓↓
Parts cost	↑	---	↓	↑

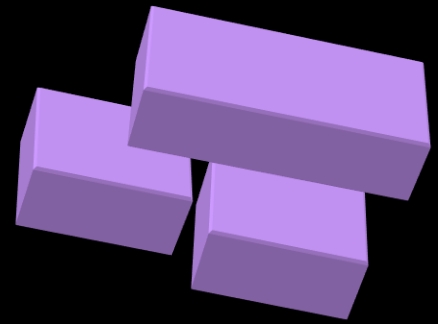
- Lego is “optimally robust” (Pareto) not “optimal.”
- Similar to complex engineering systems and biology.

↑ **better**
↓ **worse**

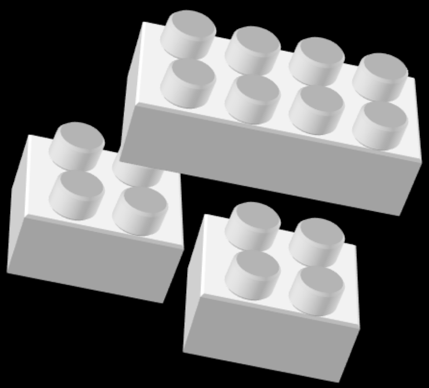


	Smooth	WT	Glue	Mold
Performance	↓	---	---	↑↑
Trauma	↓↓	---	↑	↑↑
Allowed connections	↑↑	---	---	↓↓
Reuse	---	---	↓↓	↓↓
Evolvable parts	↓	---	---	↓
Evolvable systems	↓↓	---	---	↓
Labor cost	↓	---	↓	↓↓
Parts cost	↑	---	↓	↑

Fragility: Perturbing the snap connector?

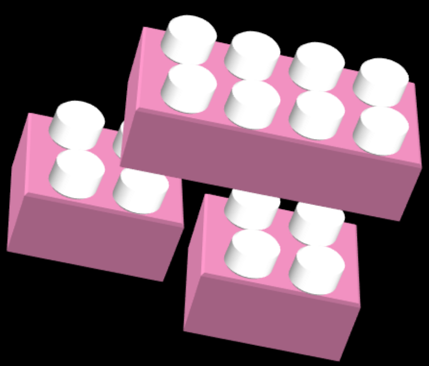


Smooth is robust



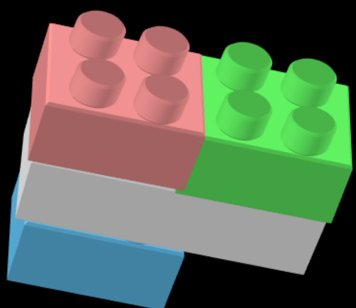
WT is ***very*** fragile

No connections, no fragility

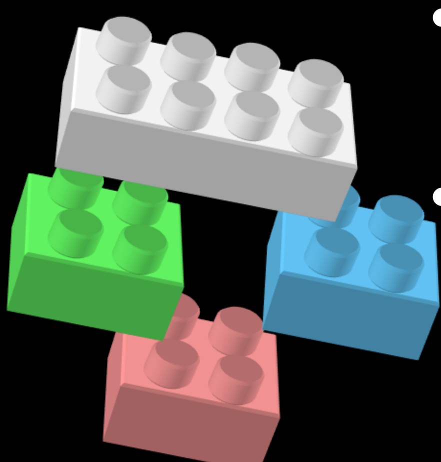


Glued is less fragile

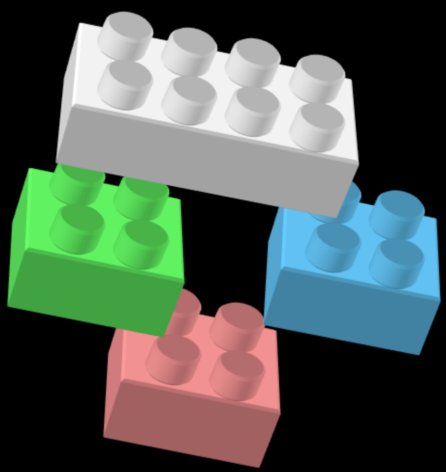
Robust or fine-tuned?

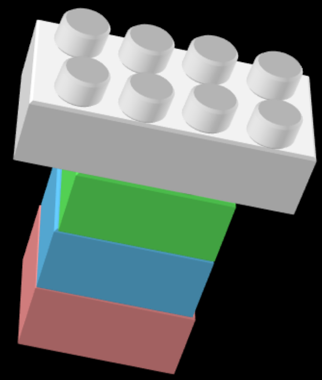


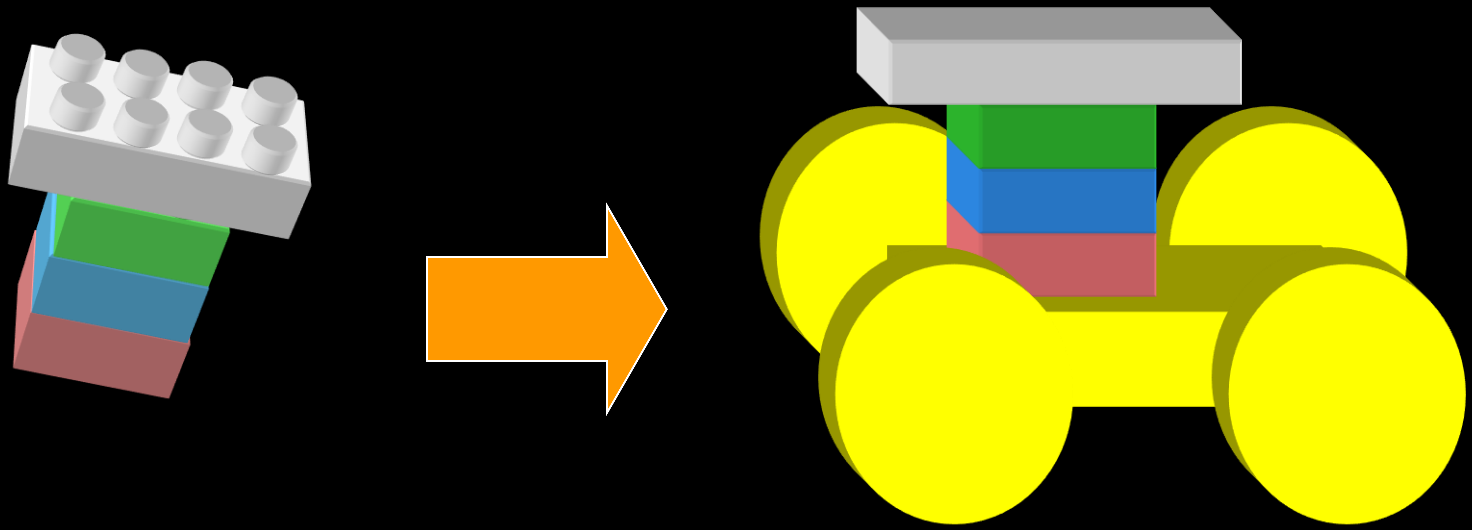
- Set of all possible interconnections is a (combinatorial) huge set.
- Set of interesting toys is also large, but an infinitesimally small subset. Very special and finely tuned.
- Similarly, among the potential toy ***systems architectures*** using the same plastic material, Lego is highly structured and finely tuned.
- At the component level, the stud-and-tube coupling is very finely machined.
- Robust yet fragile (RYF) is universal in complex engineering and biology



The evolution of complexity



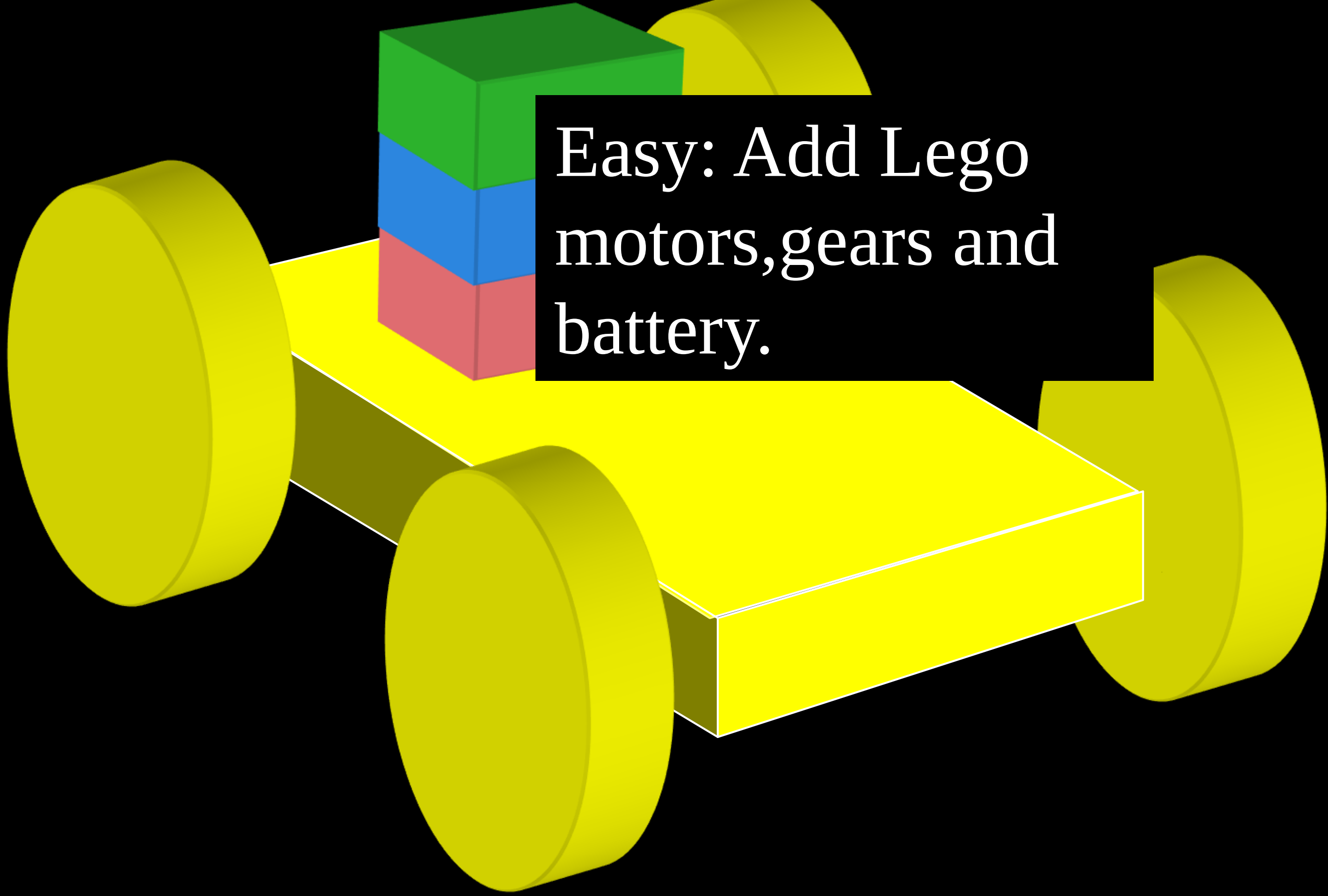




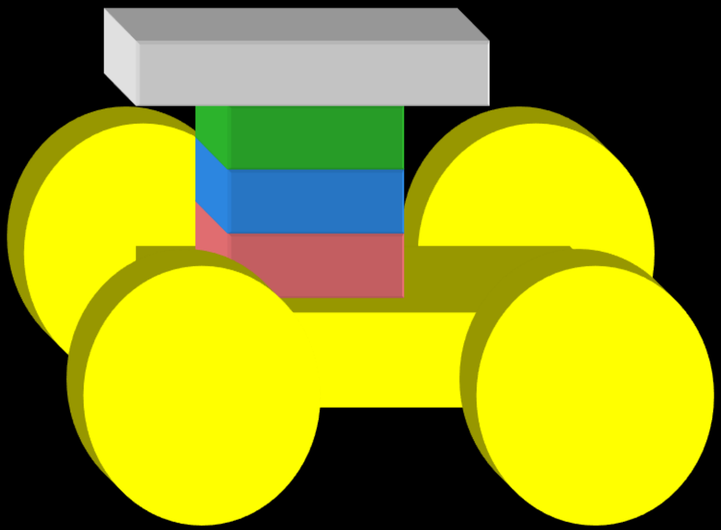
Suppose you want to put a structure on wheels?

Easy: Find Lego parts with wheels.

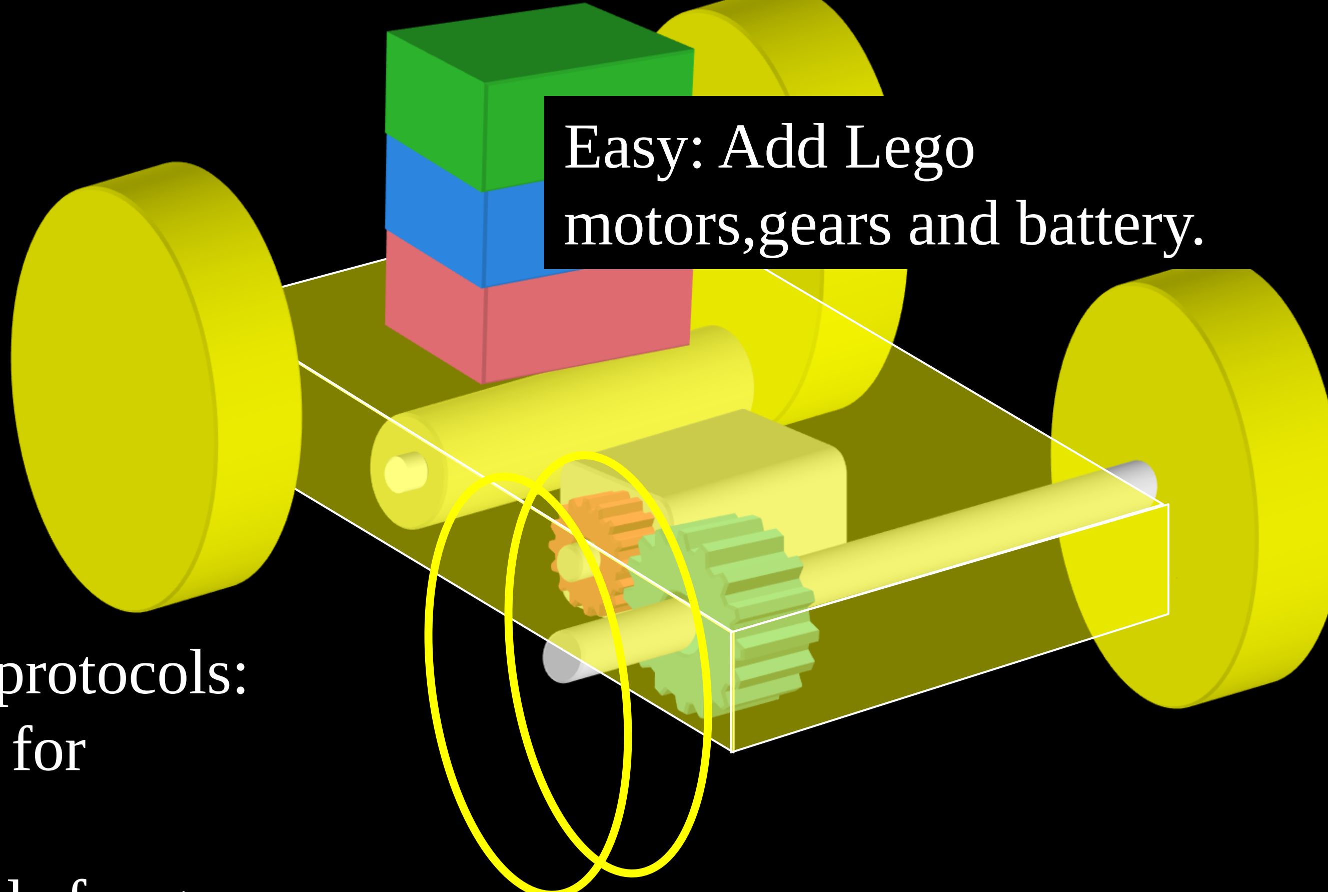
Suppose you want to motorize a vehicle with wheels?



Easy: Add Lego motors,gears and battery.



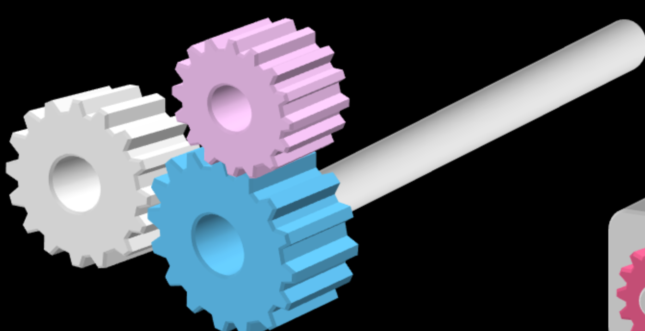
Suppose you want to motorize a vehicle with wheels?

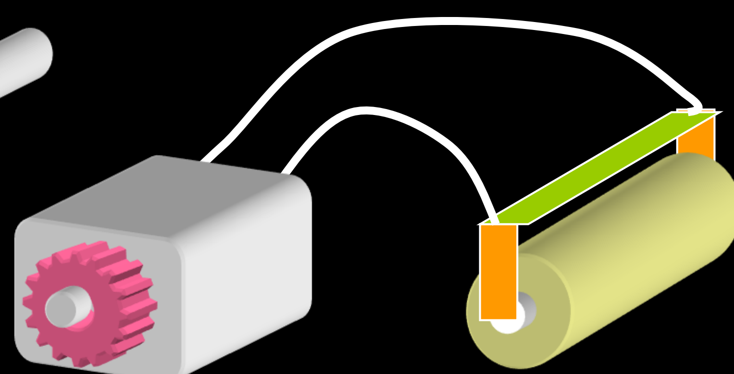


Easy: Add Lego motors,gears and battery.

This adds additional protocols:

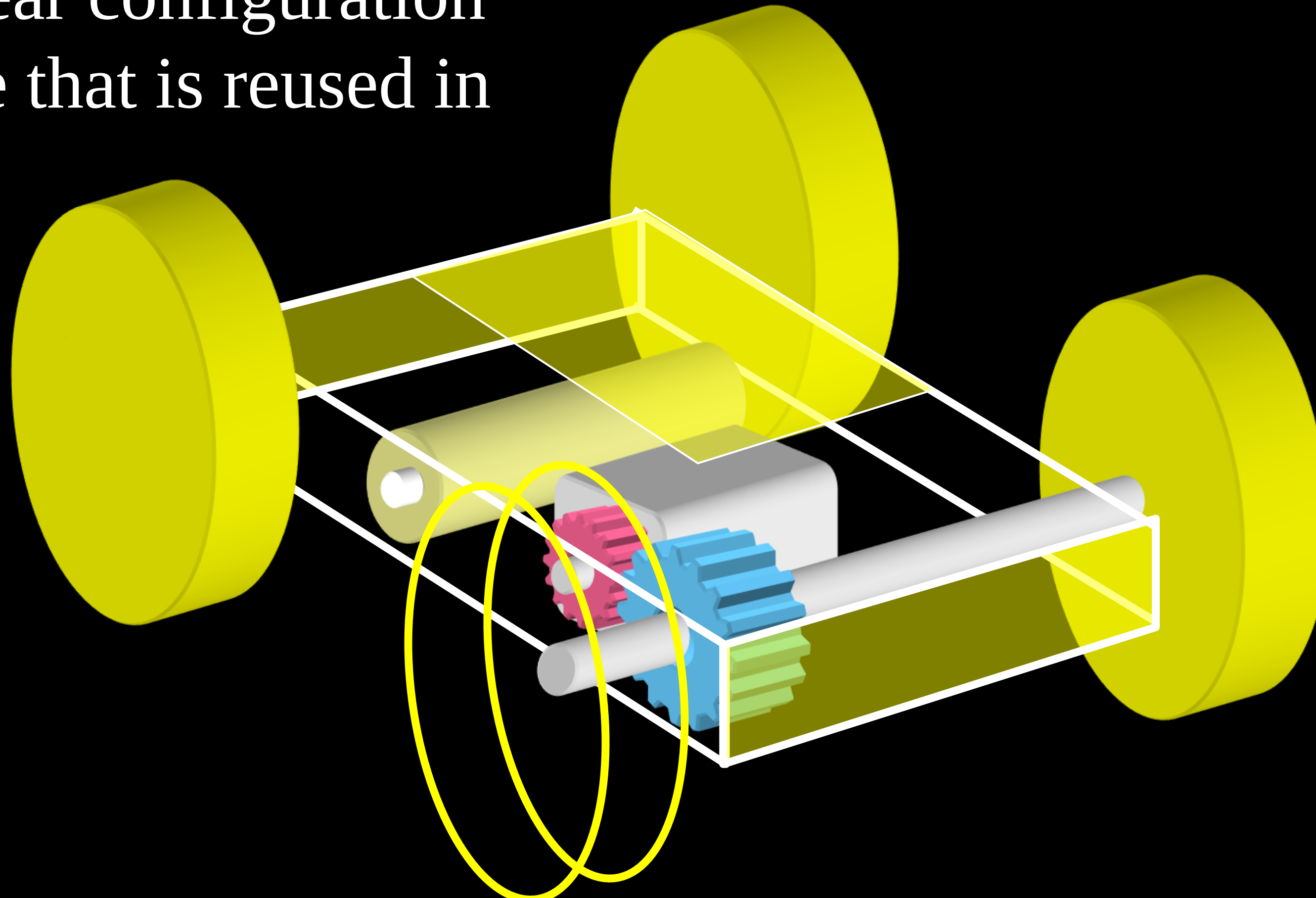
- Electrical protocols for batteries and motors.
- Mechanical protocols for gears and axles.

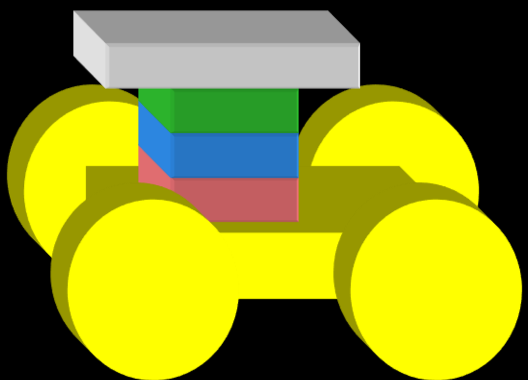


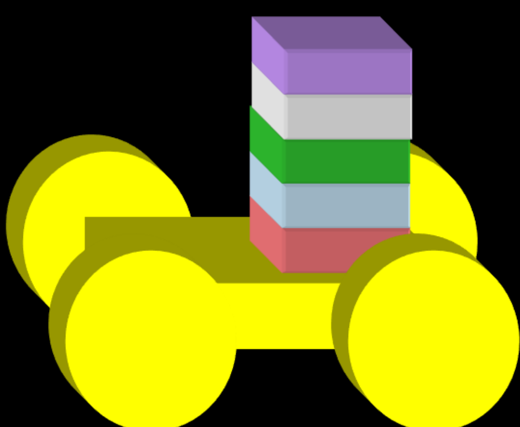


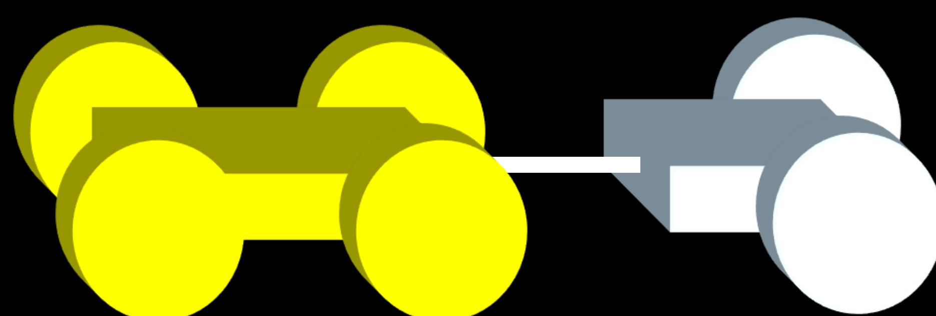
Additional protocols and modules.

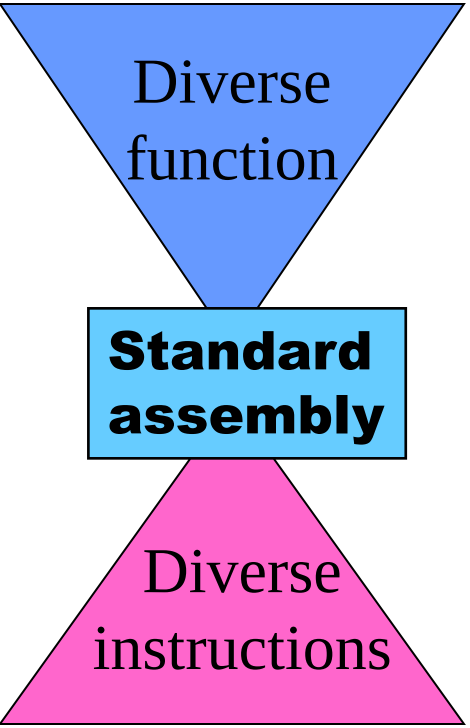
This cart/motor/gear configuration could be a module that is reused in many toys.





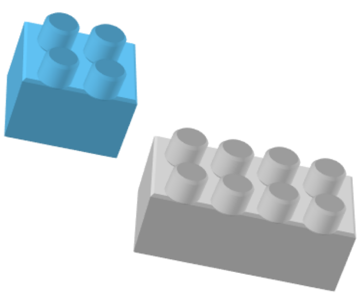


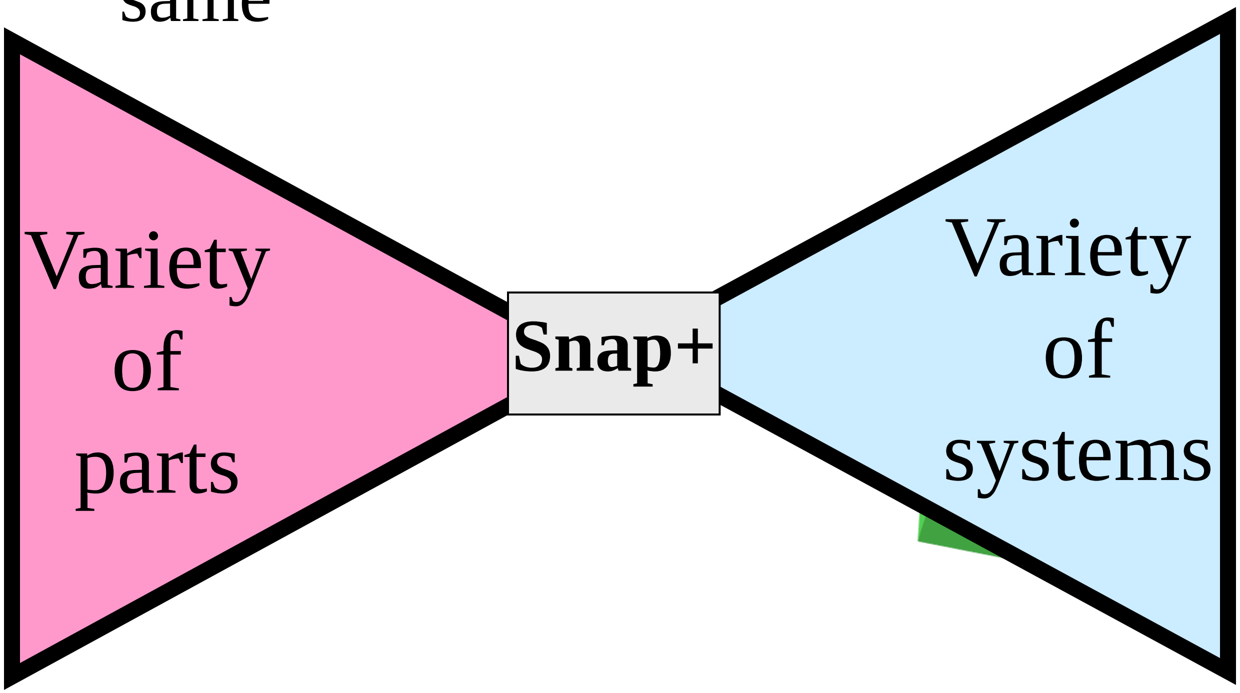




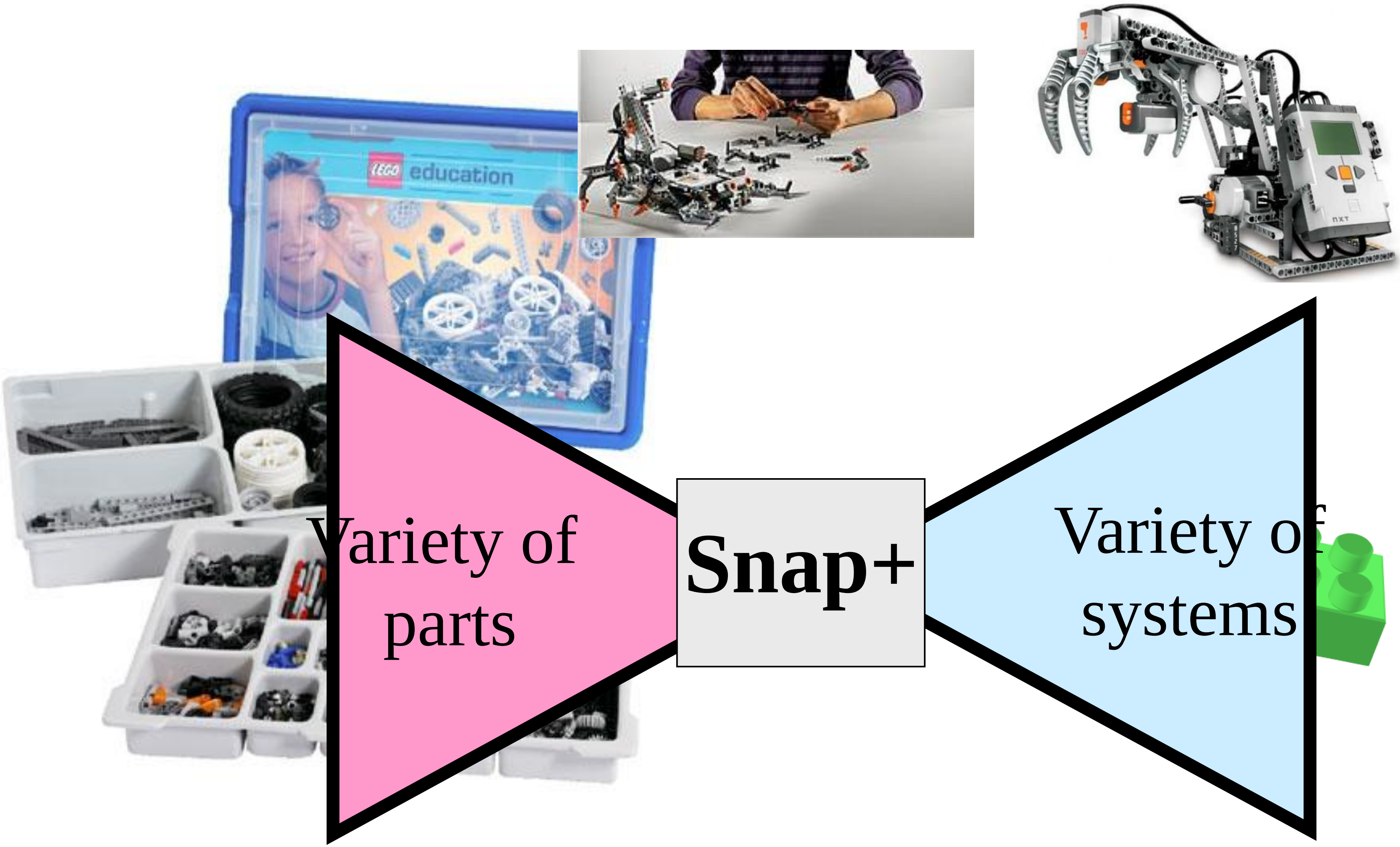
Evolvability

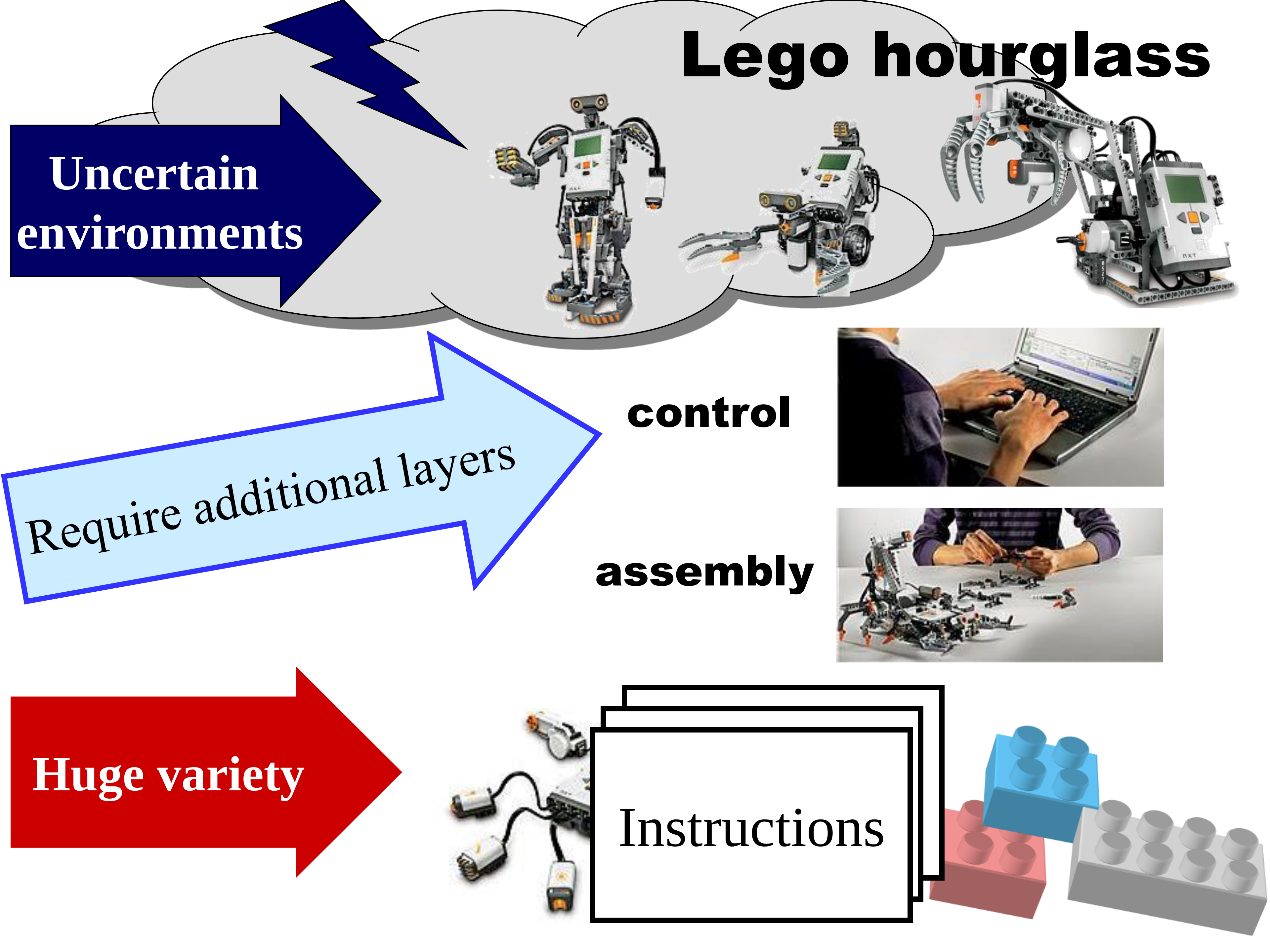
- The snap/brick can be augmented with additional parts and interfaces
- Assembly remains essentially the same



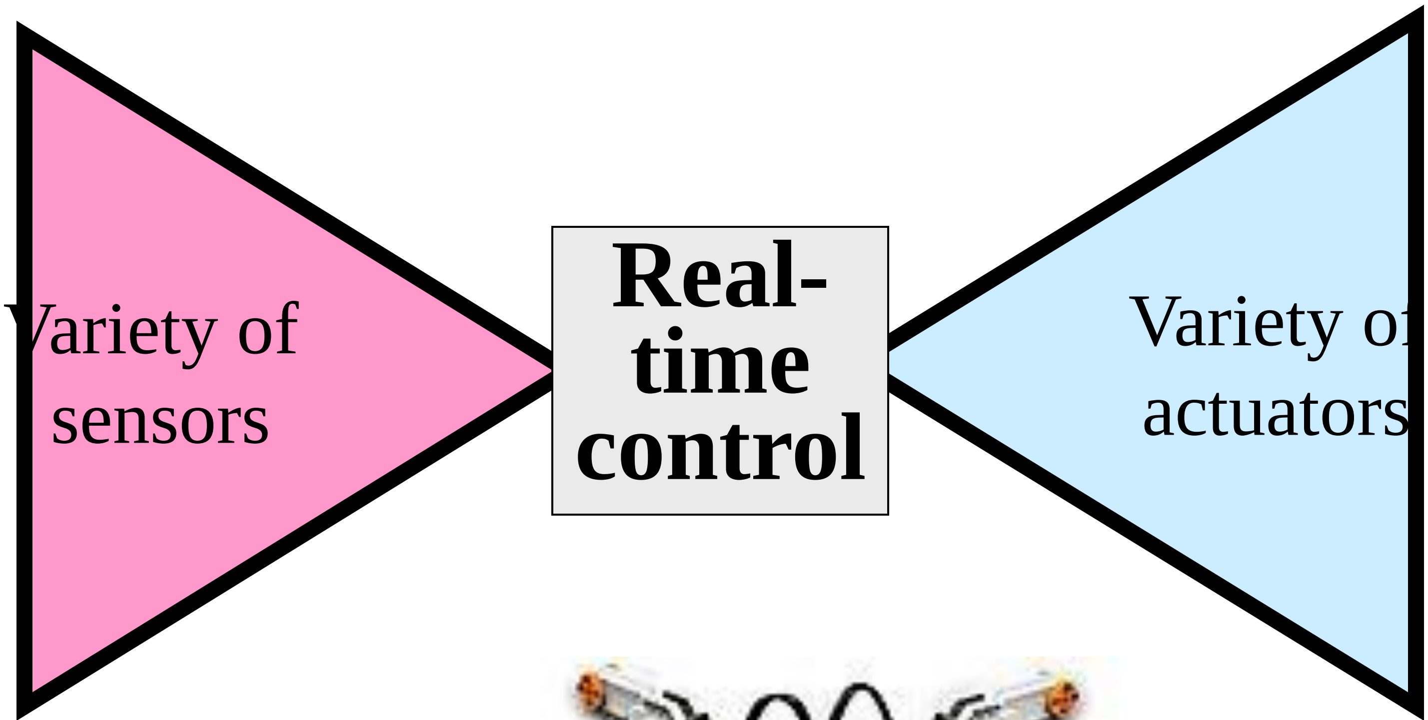


Complex toys can be created, and require additional layers of control.

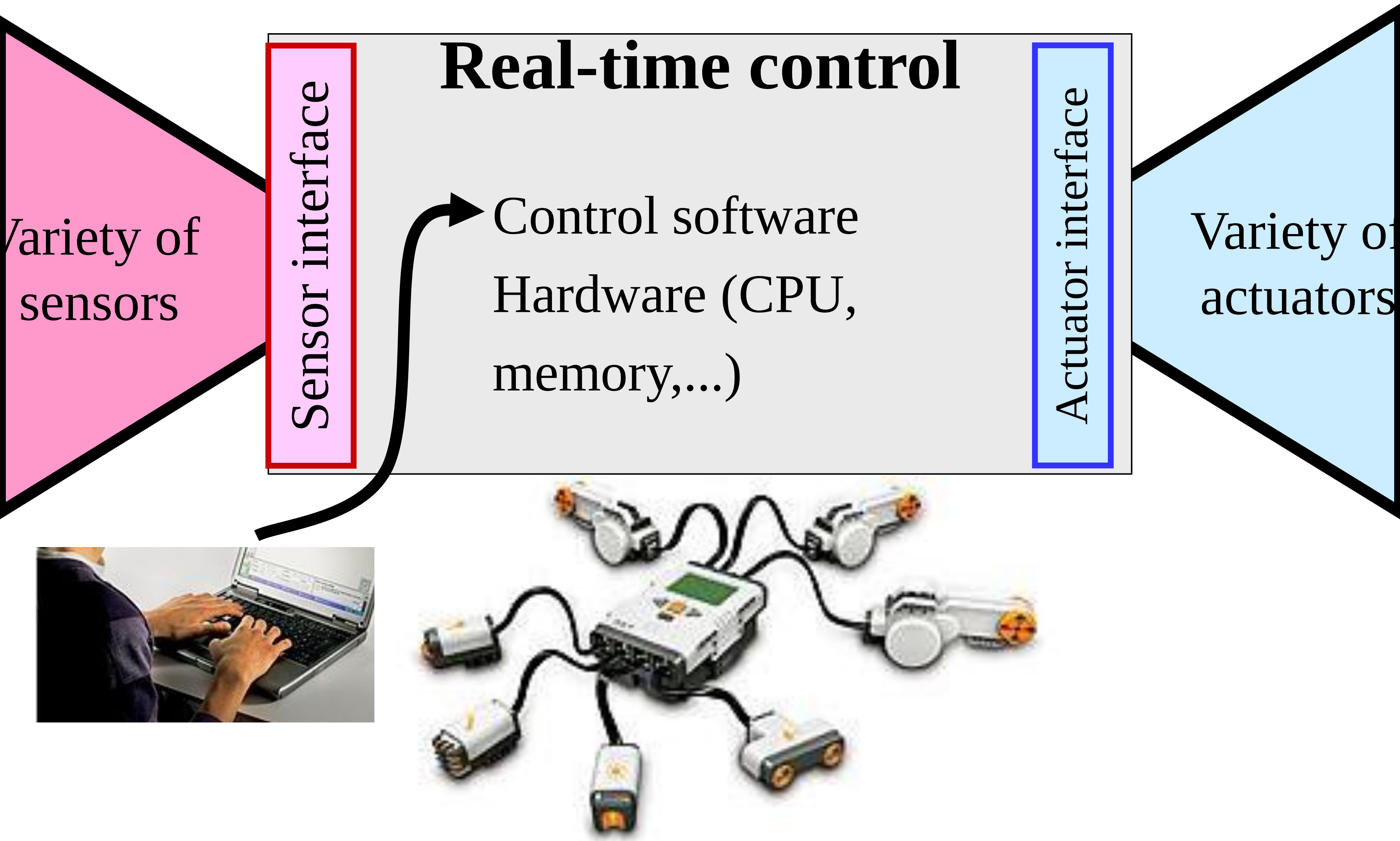


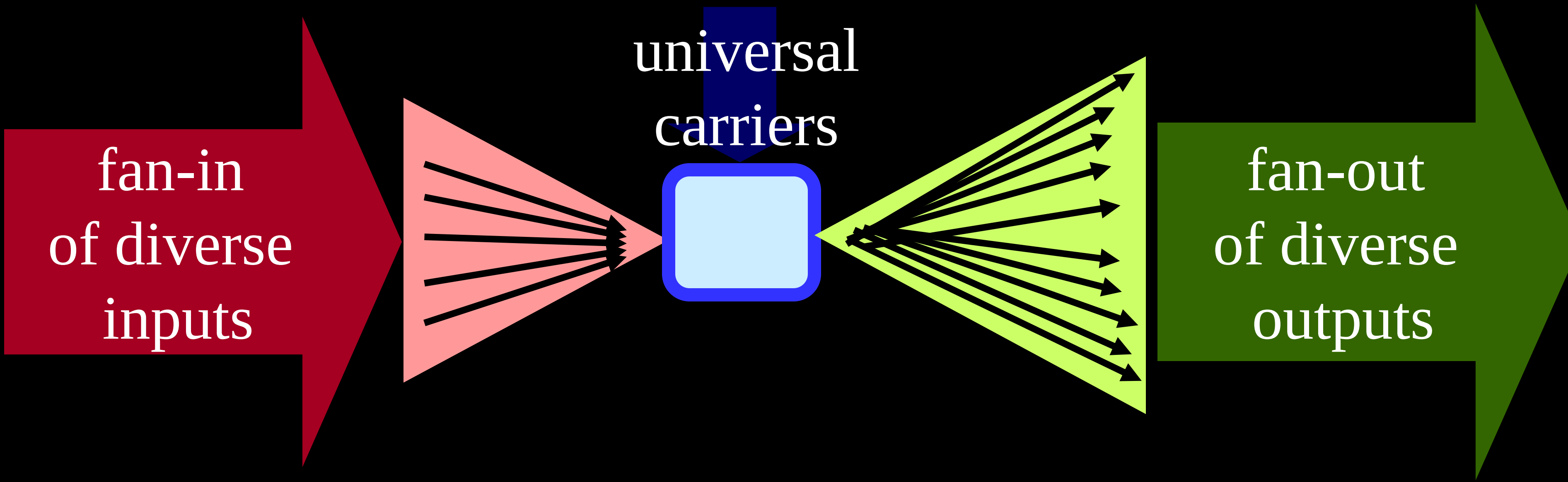

Lego hourglass
Uncertain environments
Require additional layers
control
assembly
Huge variety
Instructions

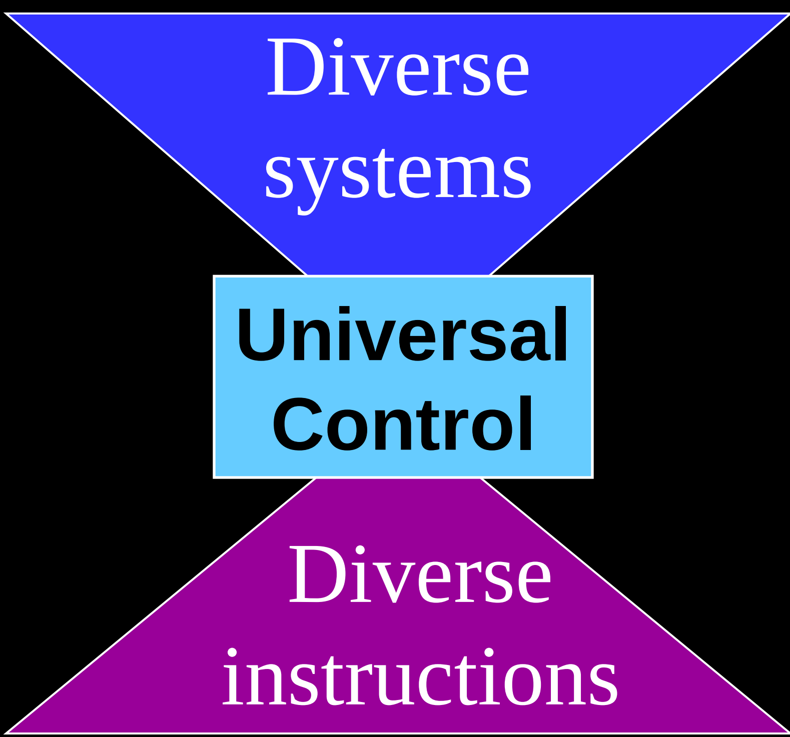
NXT controller



NXT controller







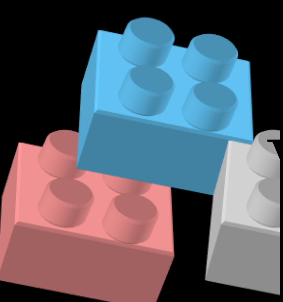
Bowties and
Hourglasses

Variety of sensors

Control

Variety of actuators

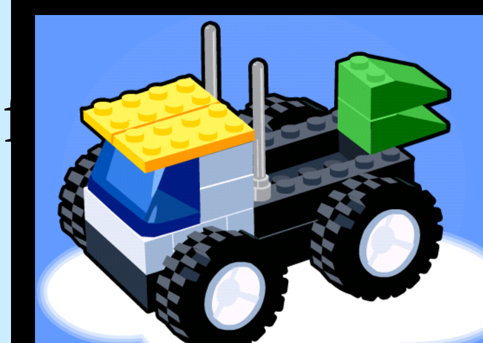
Lego Bowties



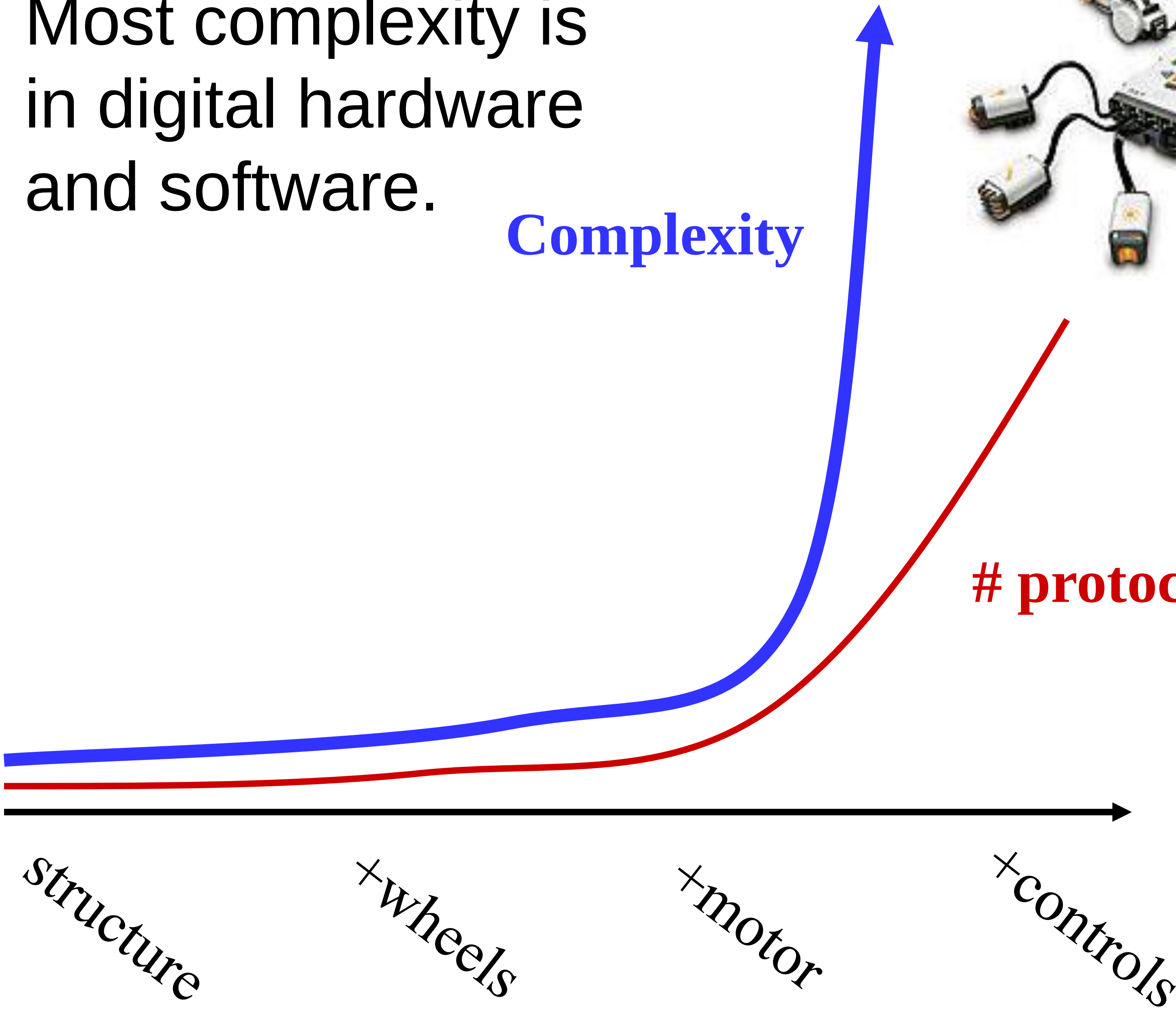
Variety of bricks

Snap

Variety of systems



Most complexity is in digital hardware and software.



Lego hourglass

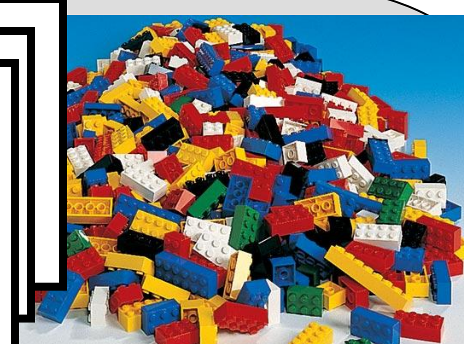
Variety

control

assembly

Variety

Instructions



Variety

Robust

Yet fragile?

Variety

Real-time control

Variety of sensors

Interface

Control software

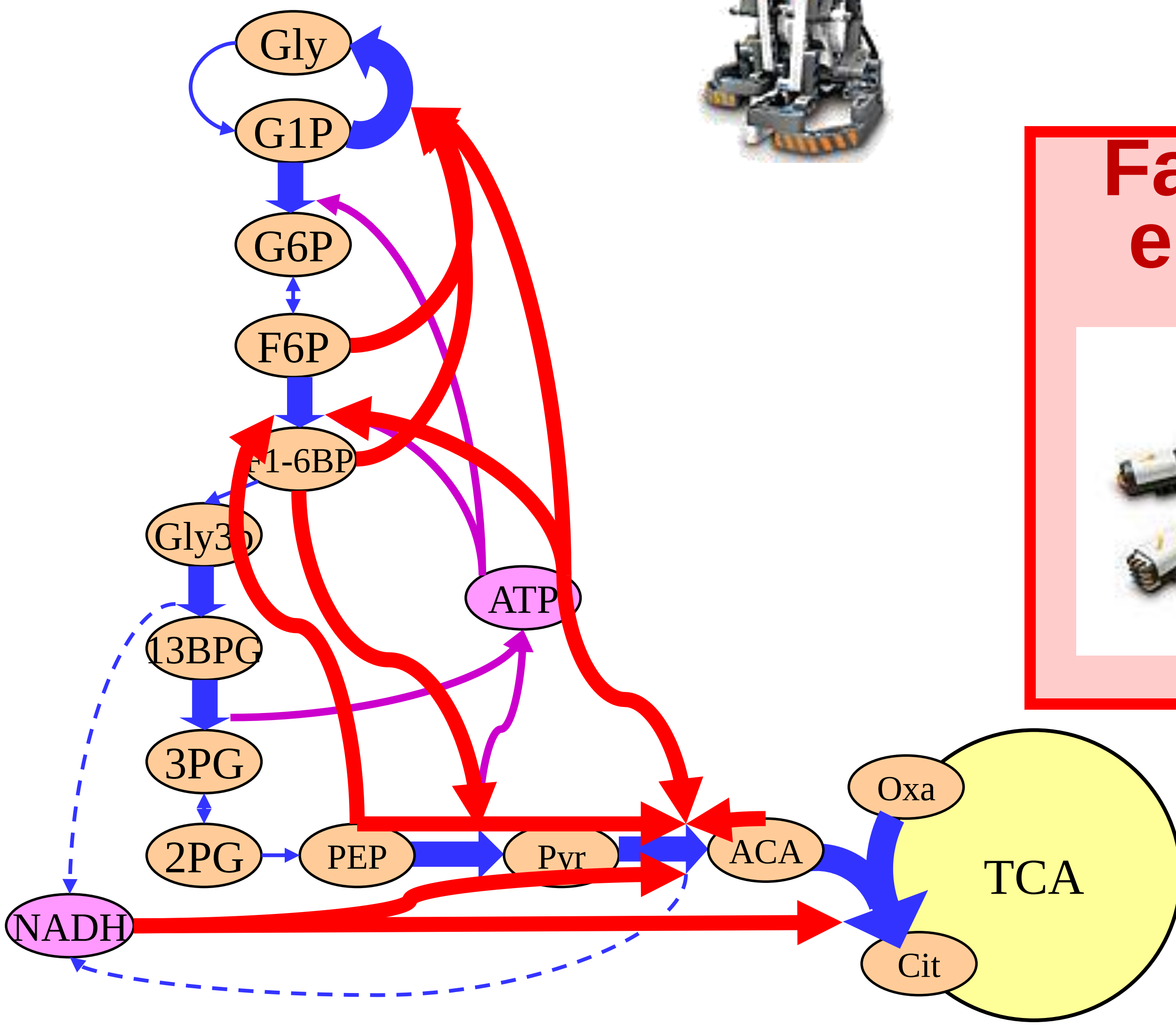
Hardware (CPU, memory,...)

Interface

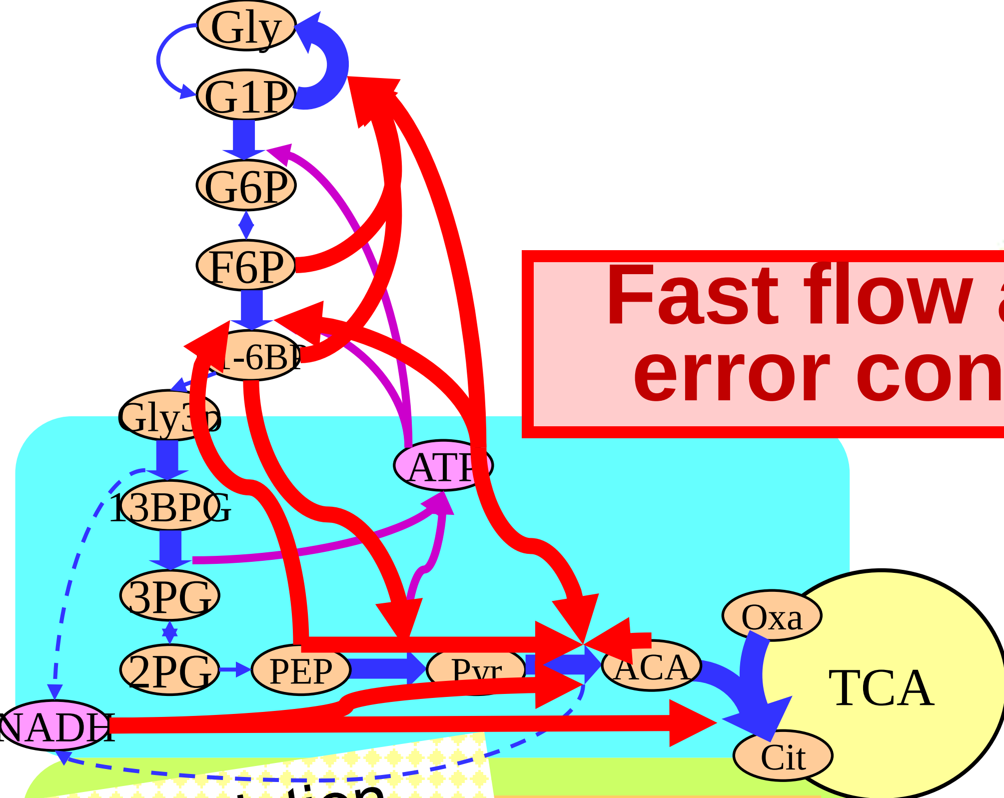
Variety of actuators

Complexity

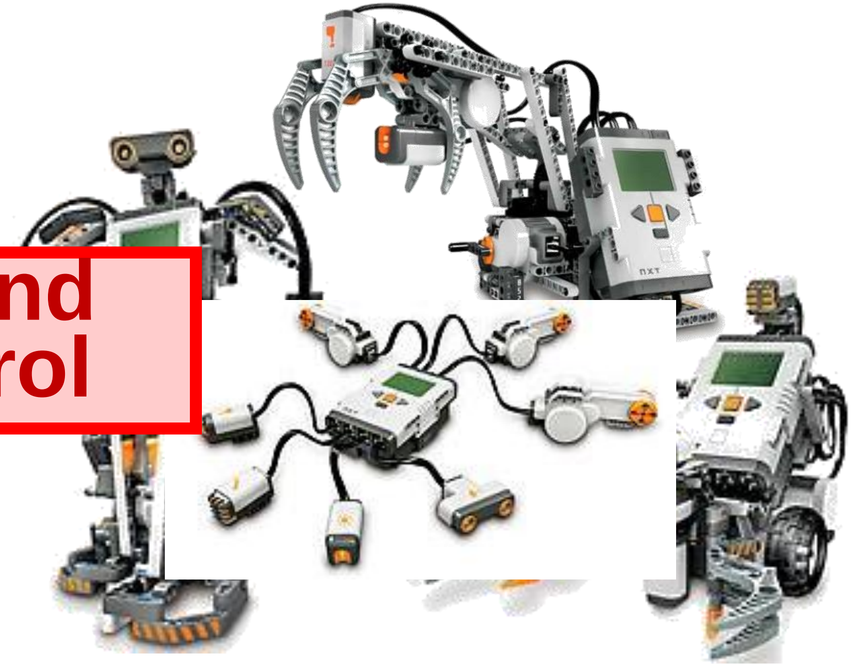
Yet fragile?



Fast flow and error control



Fast flow and error control



Translation
Reactions
Flow/error
RNA level

assembly

Transcription
Reactions
Flow/error
DNA level

Instructions

For a single toy

Lego hourglass

Complexity

control

assembly

Toy *system*

Lego hourglass

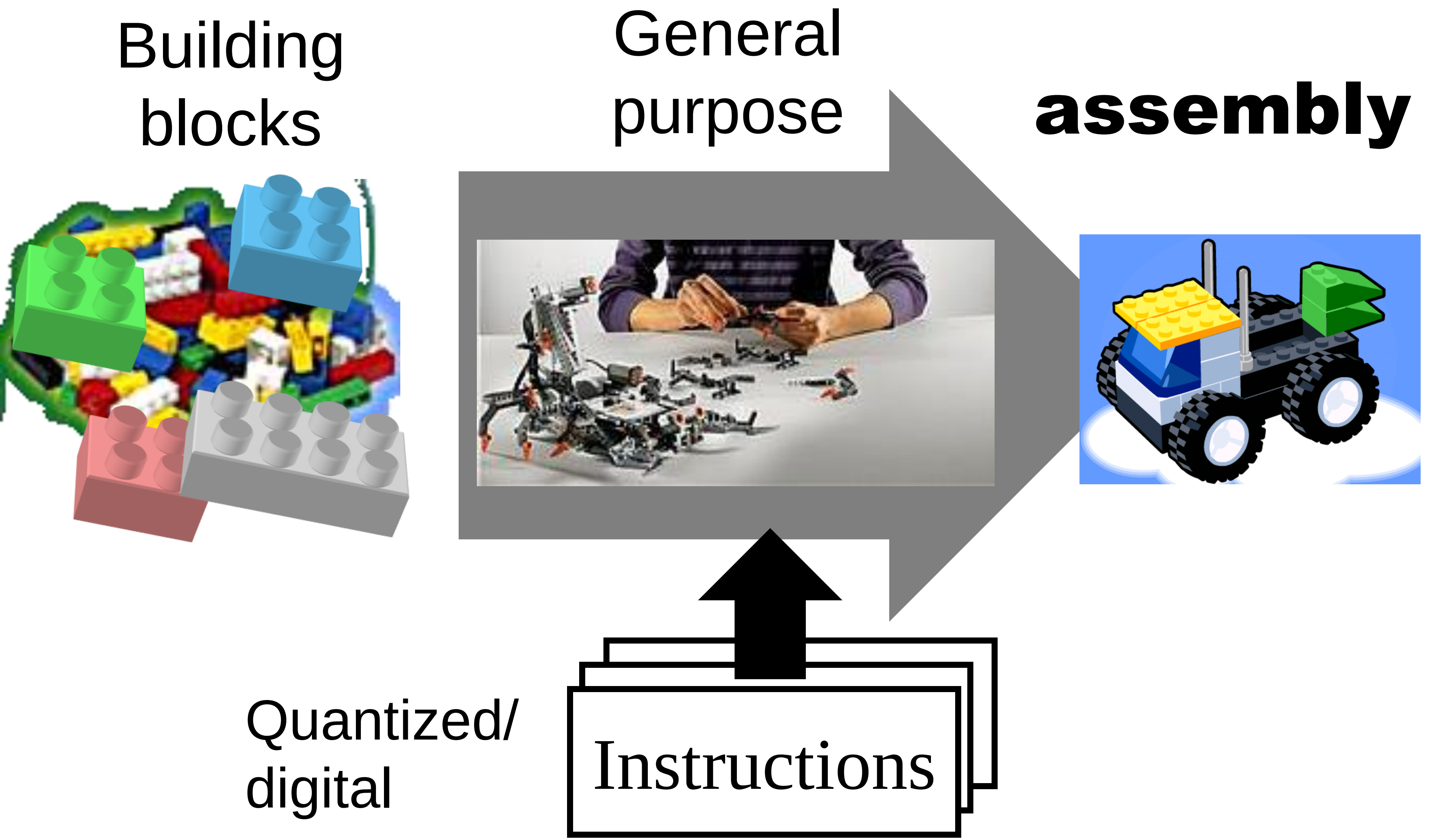
Huge variety of toys

Standardized mechanisms
Highly conserved

control

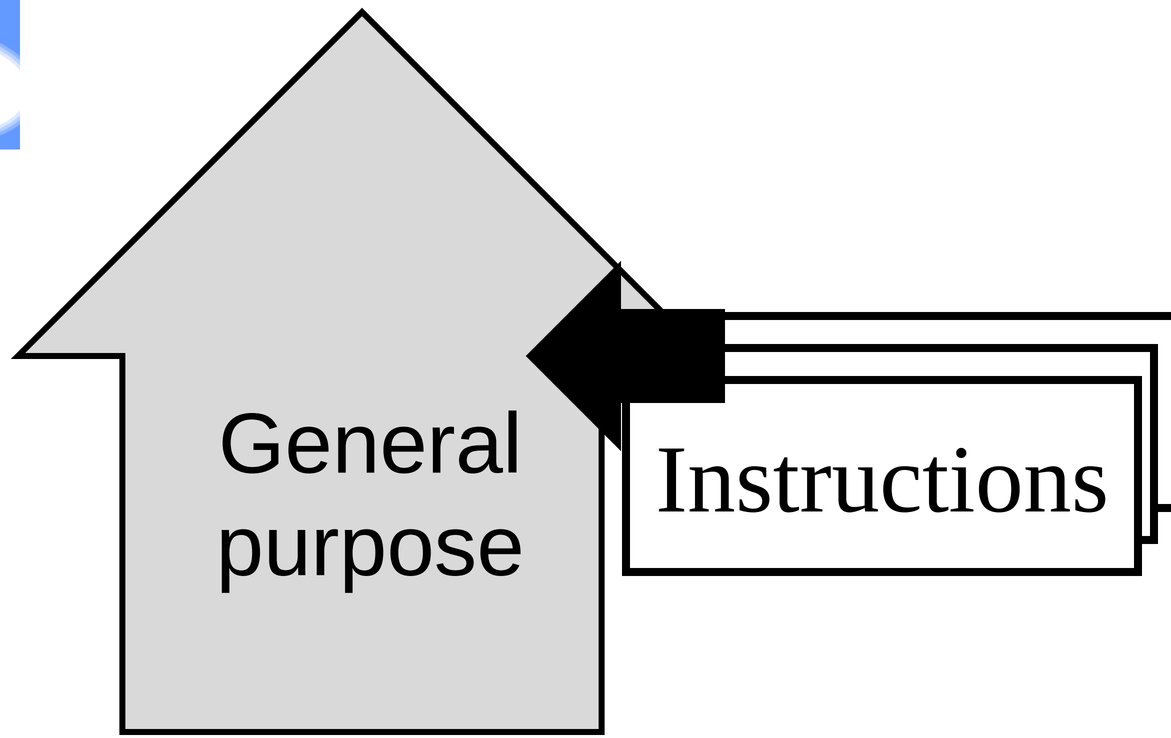
assembly

Huge variety of instructions


Building
blocks
General
purpose
assembly
Quantized/
digital
Instructions

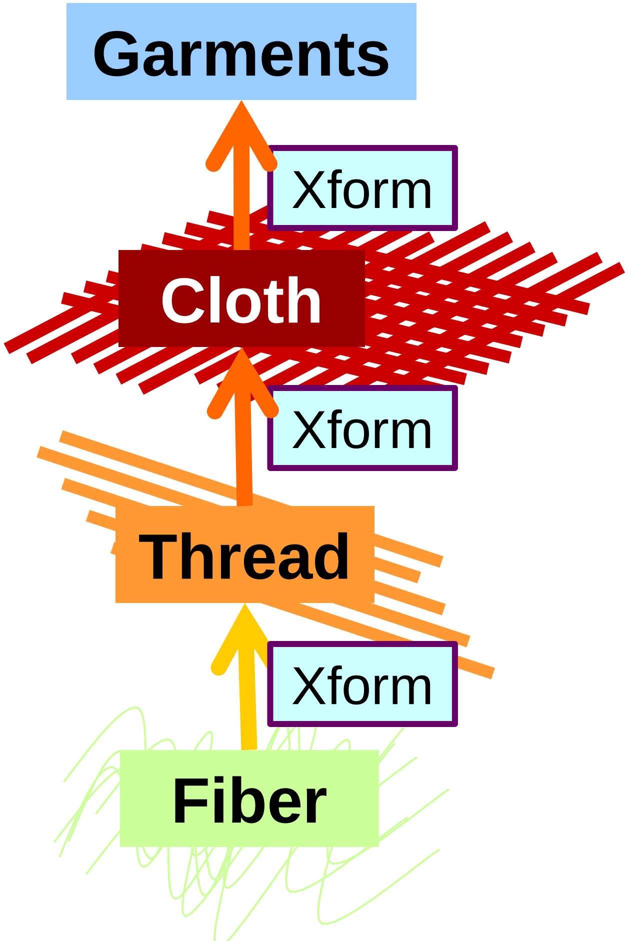
Each step uses general purpose machines

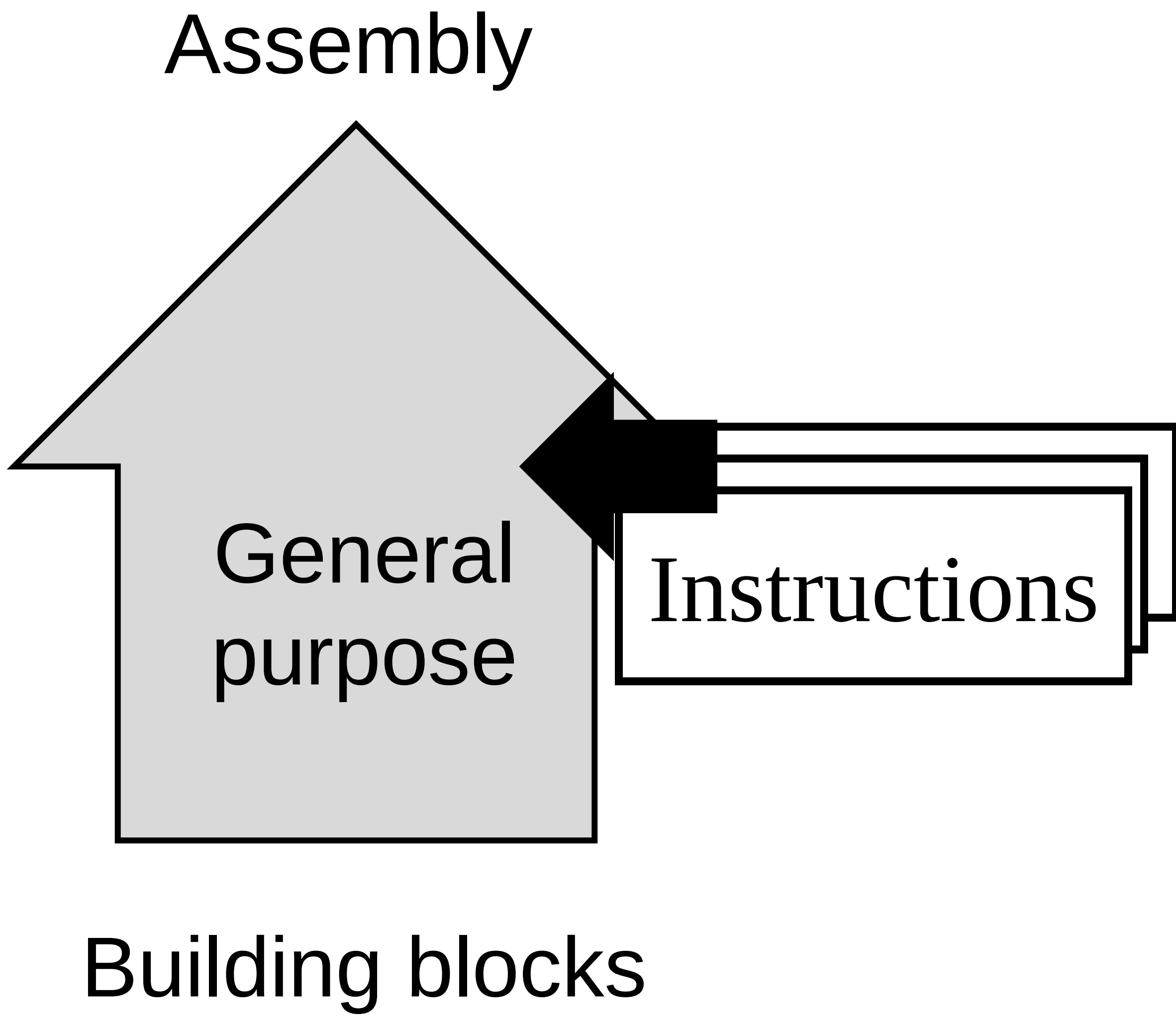
Assembly



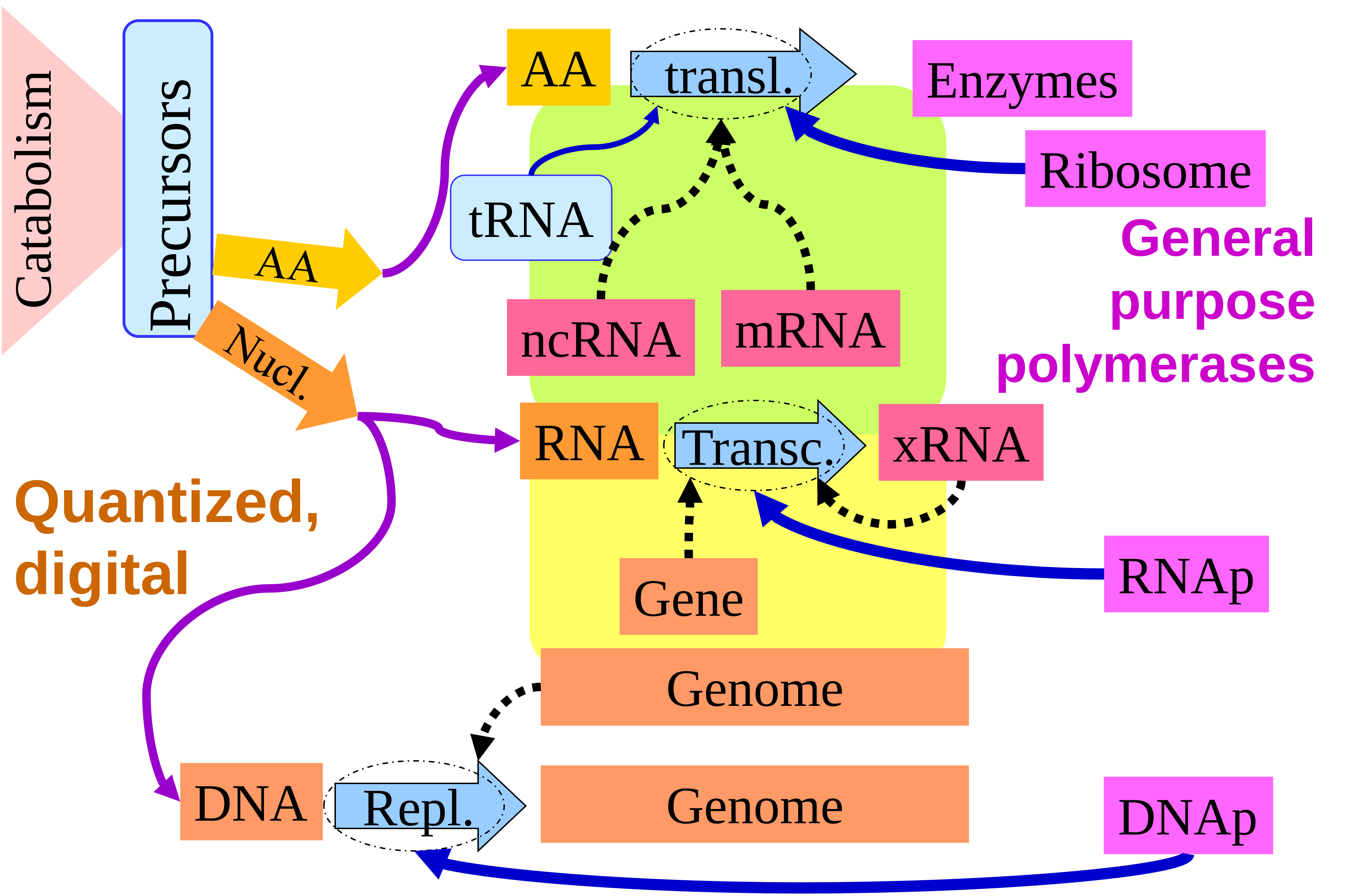
Building blocks

Each step uses general purpose machines

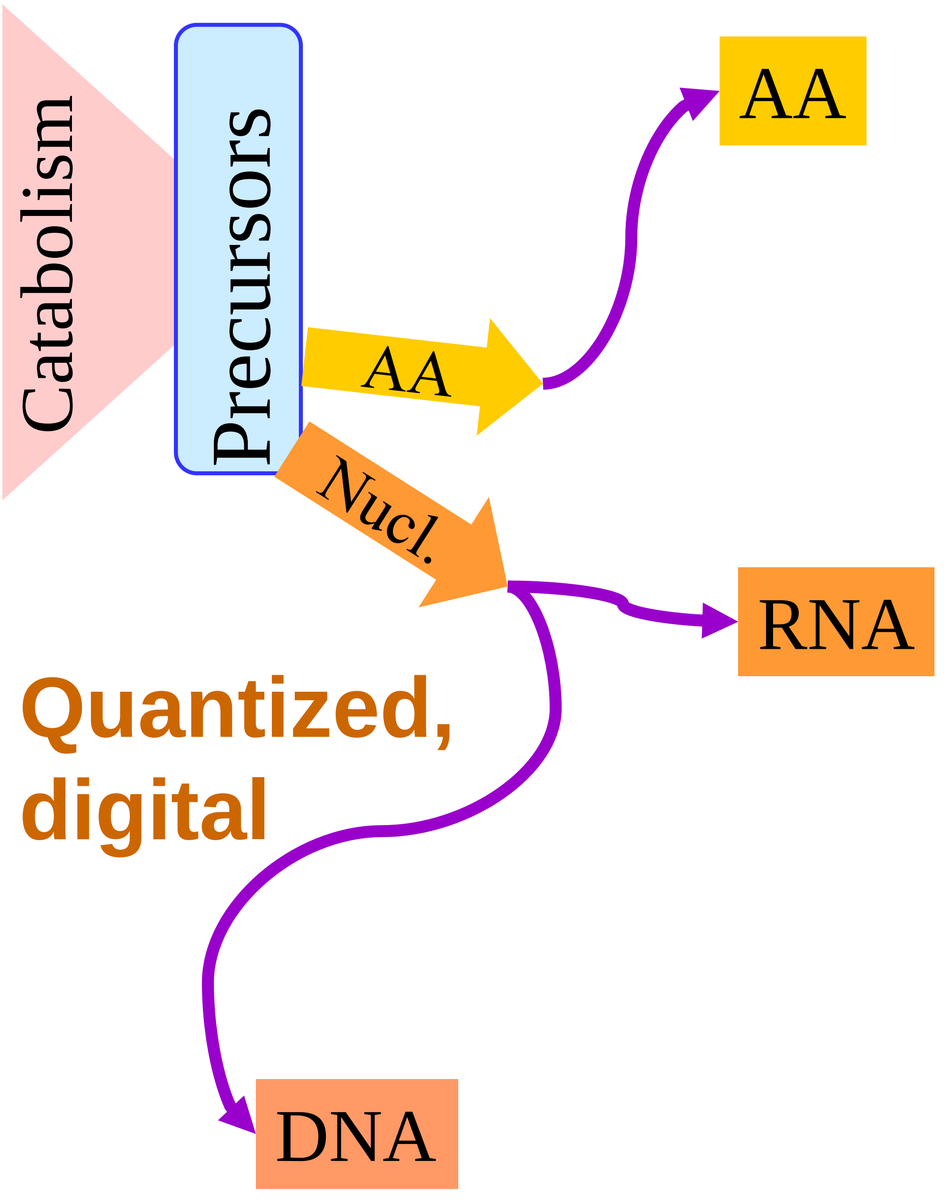




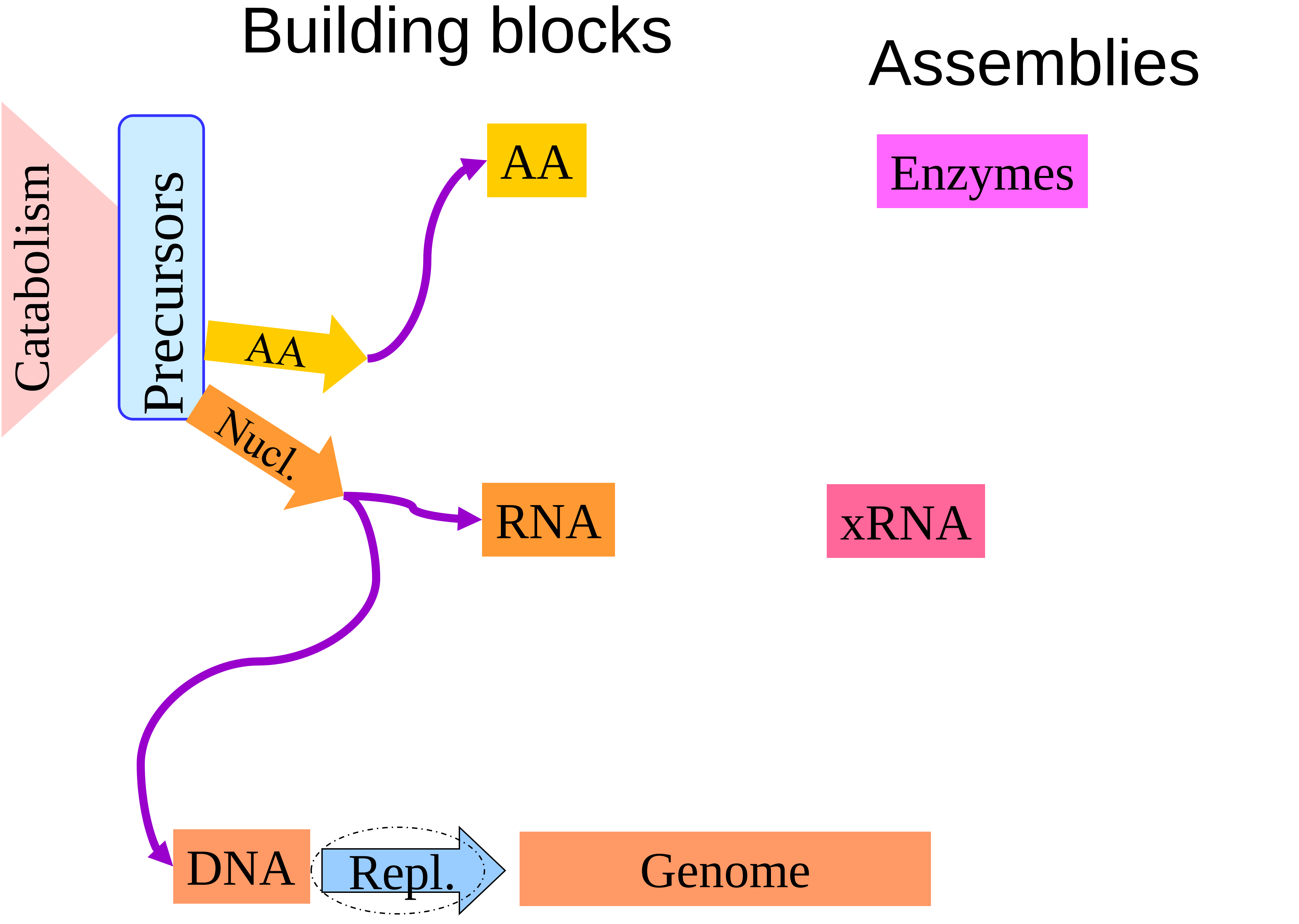
Lower layers

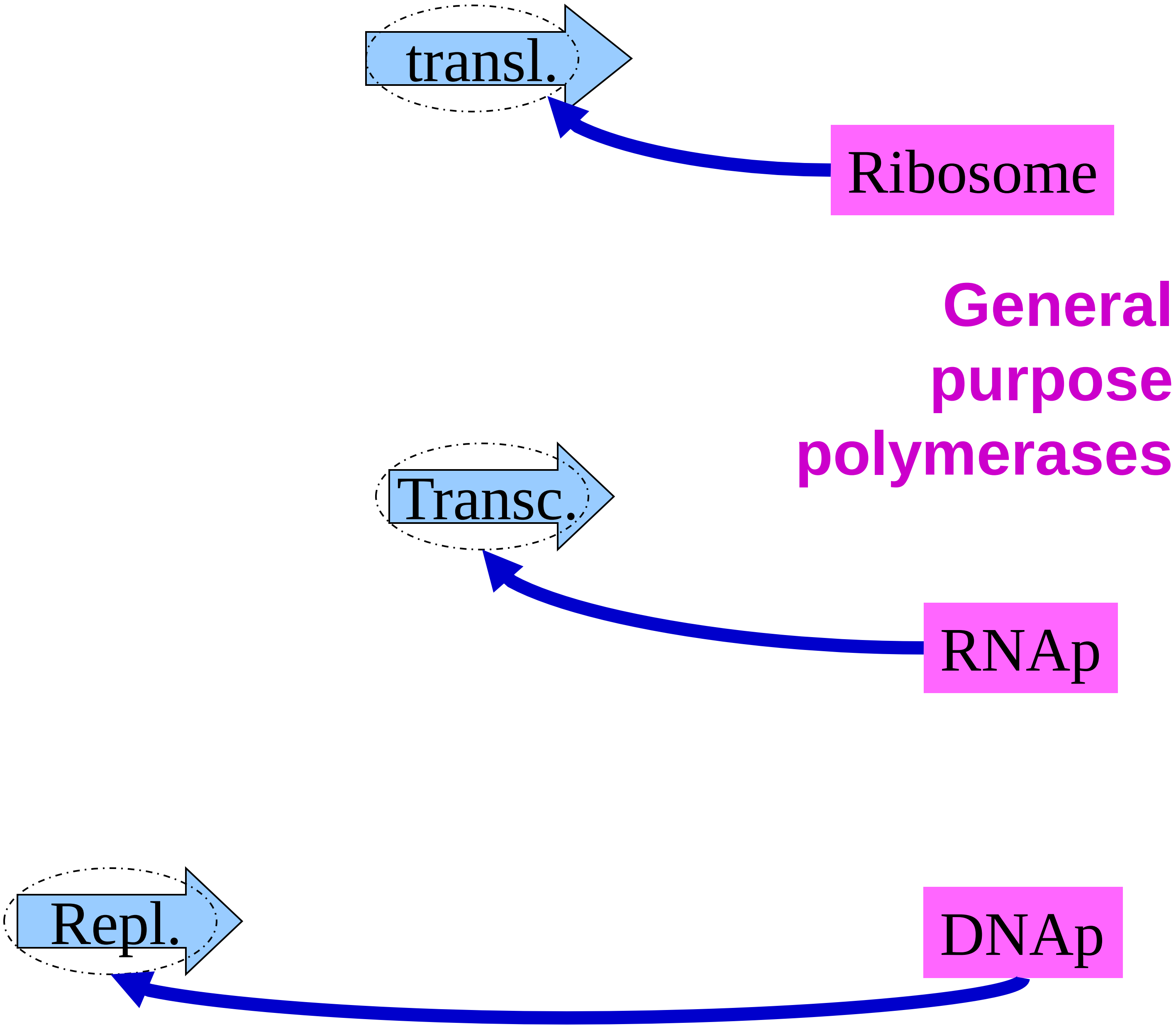


Lower layers

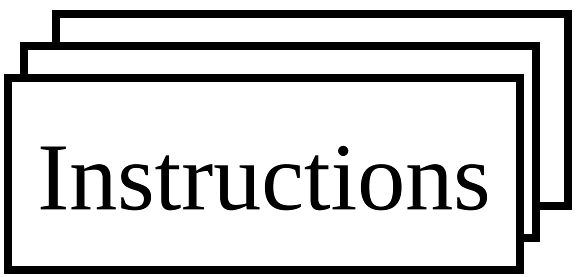


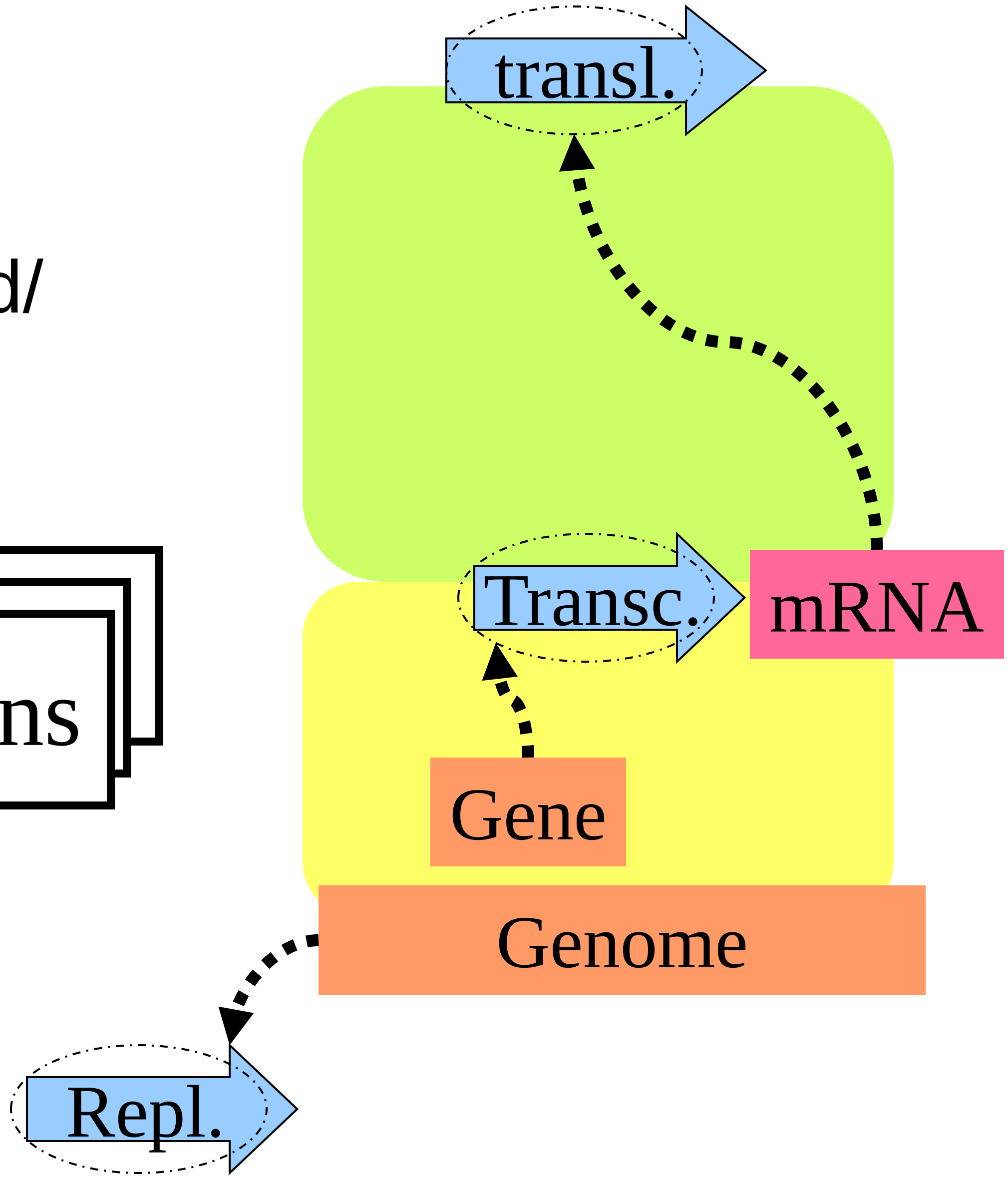
Building blocks


Building blocks
Assemblies
Catabolism
Precursors
AA
Nucl.
AA
RNA
DNA
Repl.
Genome
Enzymes
xRNA

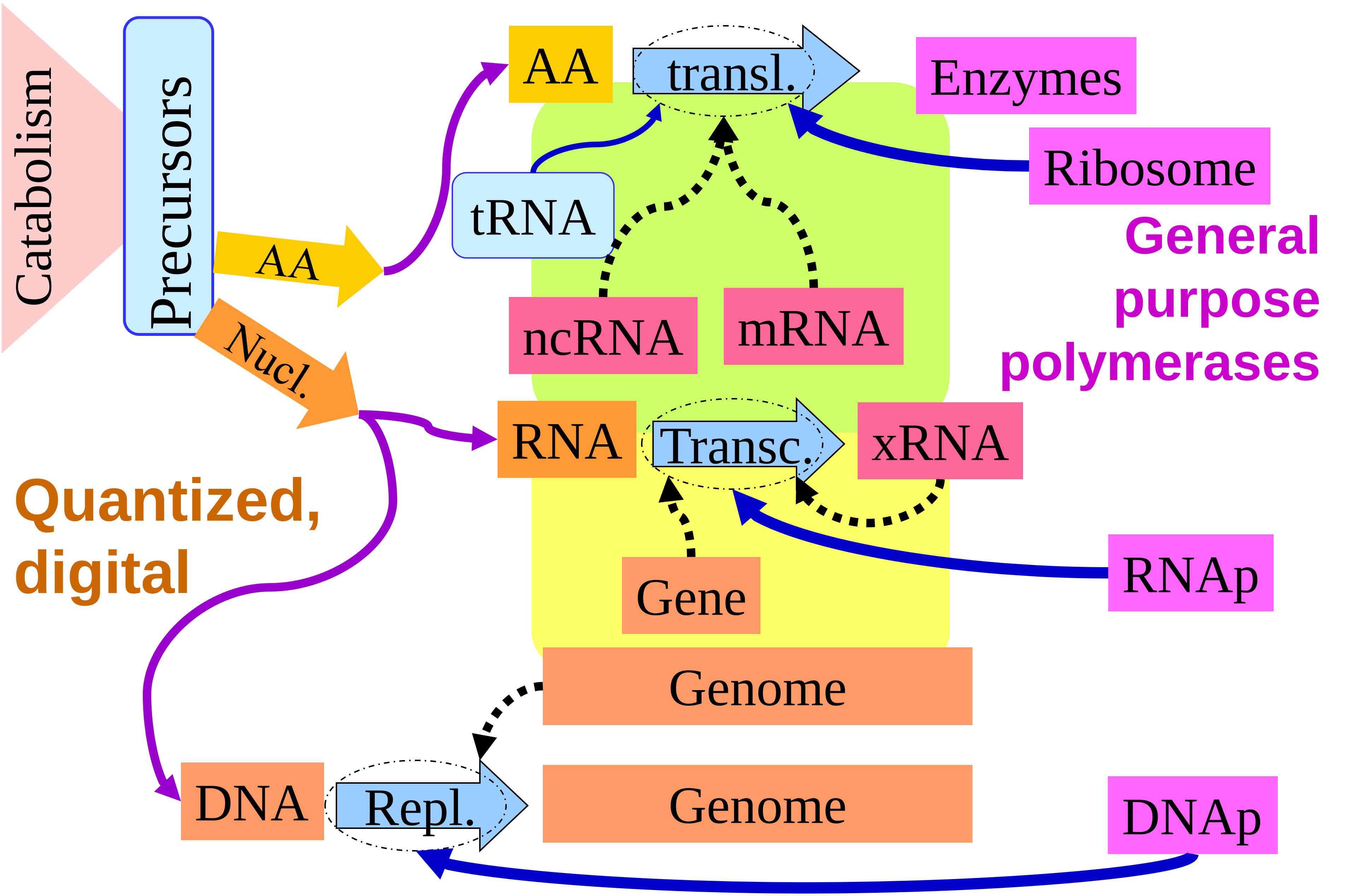

transl.
Ribosome
General
purpose
polymerases
Transc.
RNAp
Repl.
DNAp

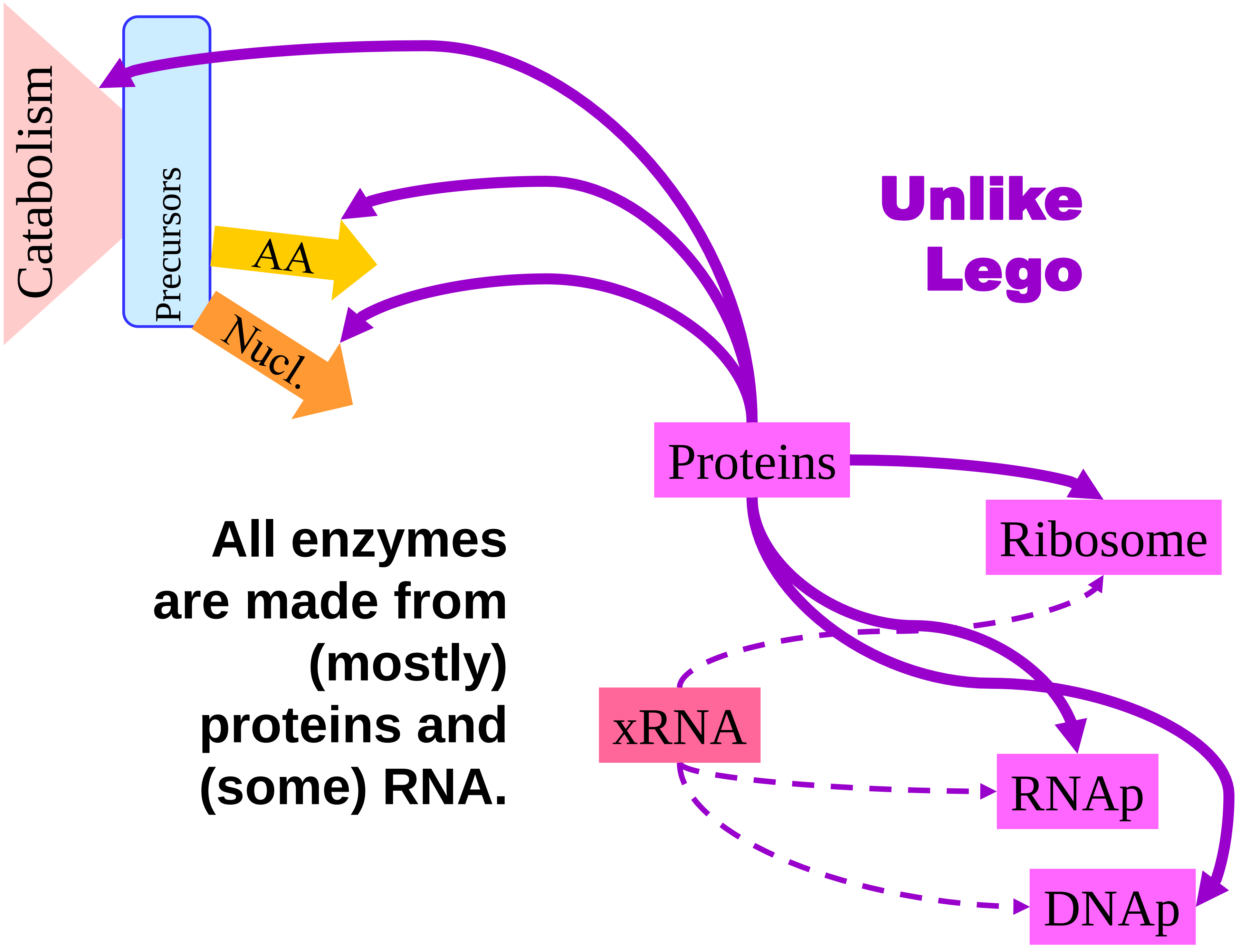
Quantized/
digital



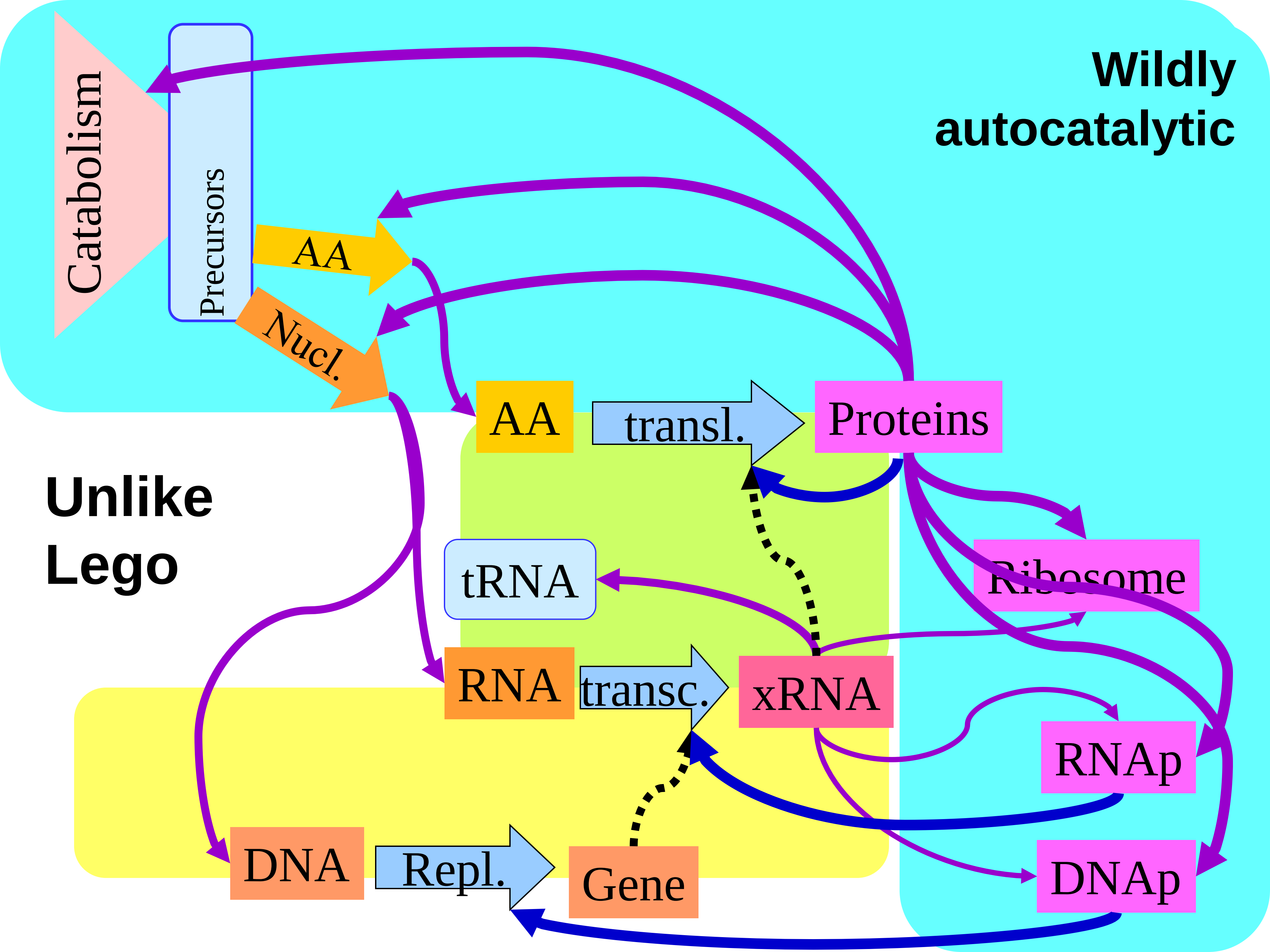


Lower layers





All enzymes are made from (mostly) proteins and (some) RNA.



Wildly
autocatalytic
Catabolism
Precursors
AA
Nucl.
AA
transl.
Proteins
Unlike
Lego
tRNA
Ribosome
RNA
transc.
xRNA
RNAp
DNA
Repl.
Gene
DNAp

Lego hourglass

Huge variety of toys

Standardized mechanisms Highly conserv

control

sembly

Large (<< huge) Variety of parts

Huge variety of instructions

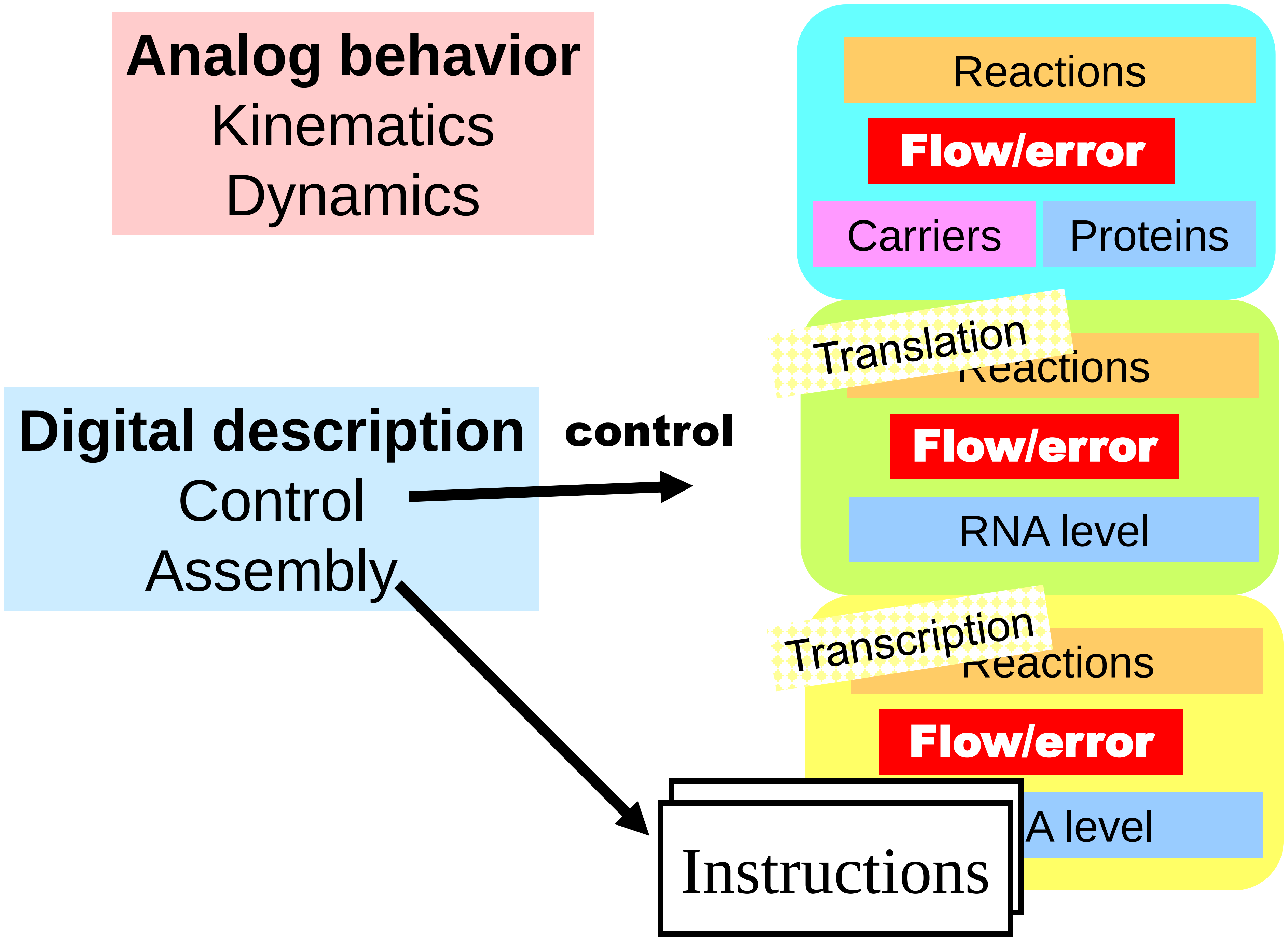
Instructions

Analog behavior
Kinematics
Dynamics

control

Digital description
Control
Assembly

Instructions

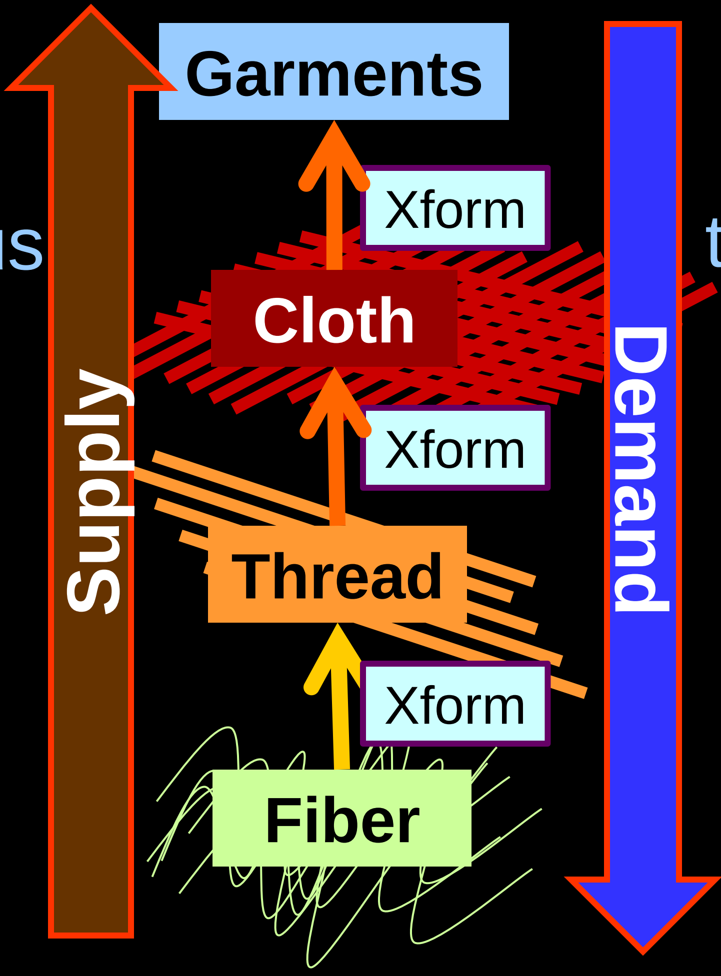


Analog behavior
Kinematics
Dynamics
Reactions
Flow/error
Carriers
Proteins
Translation
Reactions
Digital description
Control
Assembly
control
Flow/error
RNA level
Transcription
Reactions
Flow/error
A level
Instructions

Universal strategies?

Even though garments seem analog/continuous

quantization for robustness



Garments have limited access to threads and fibers

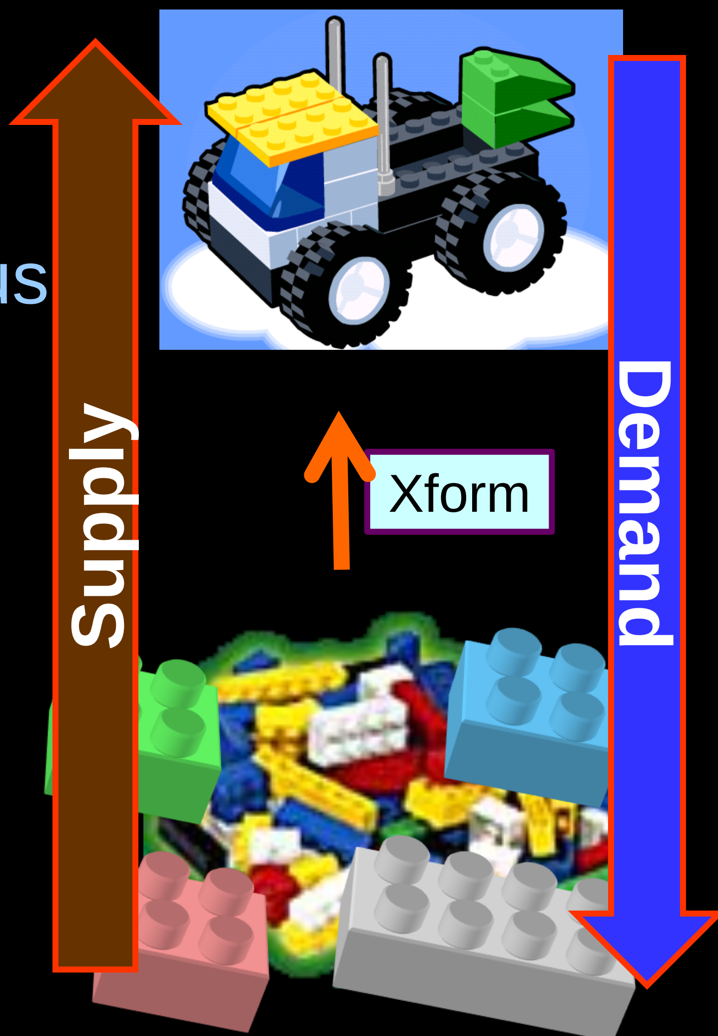
constraints on cross-layer interactions

Prevents unraveling of lower layers

Universal strategies?

Even though toys seem analog/continuous

Toys have limited access to (hidden) snaps



quantization for robustness

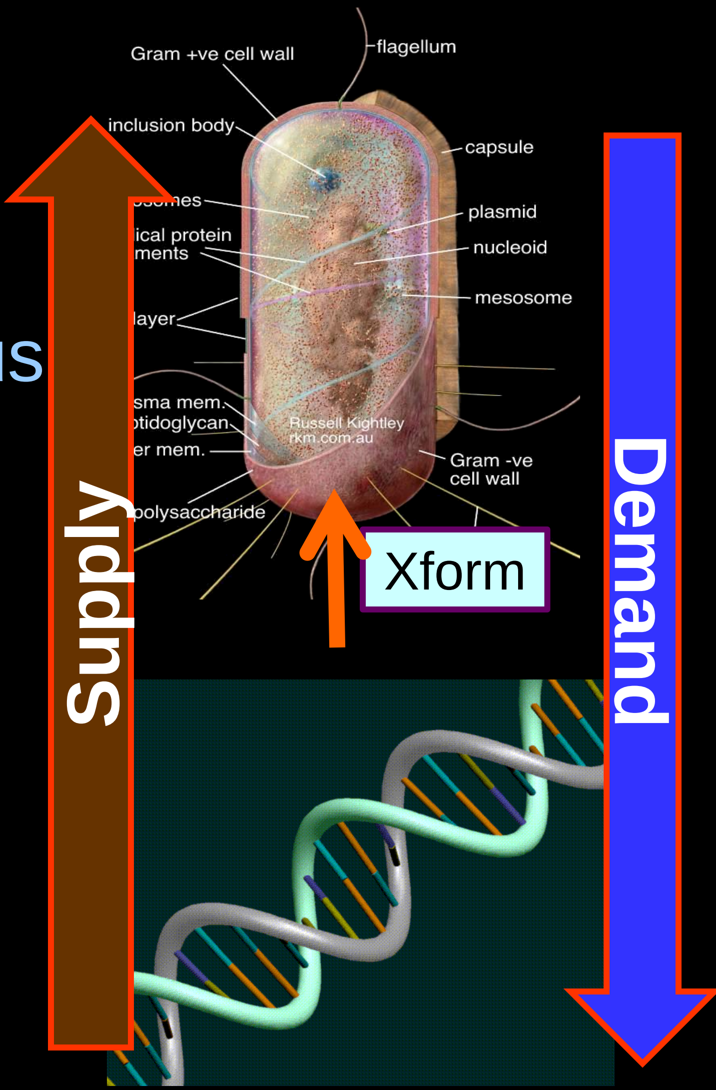
constraints on cross-layer interactions

Prevents unraveling of lower layers

Universal strategies?

Even though
Cells seem
analog/continuous

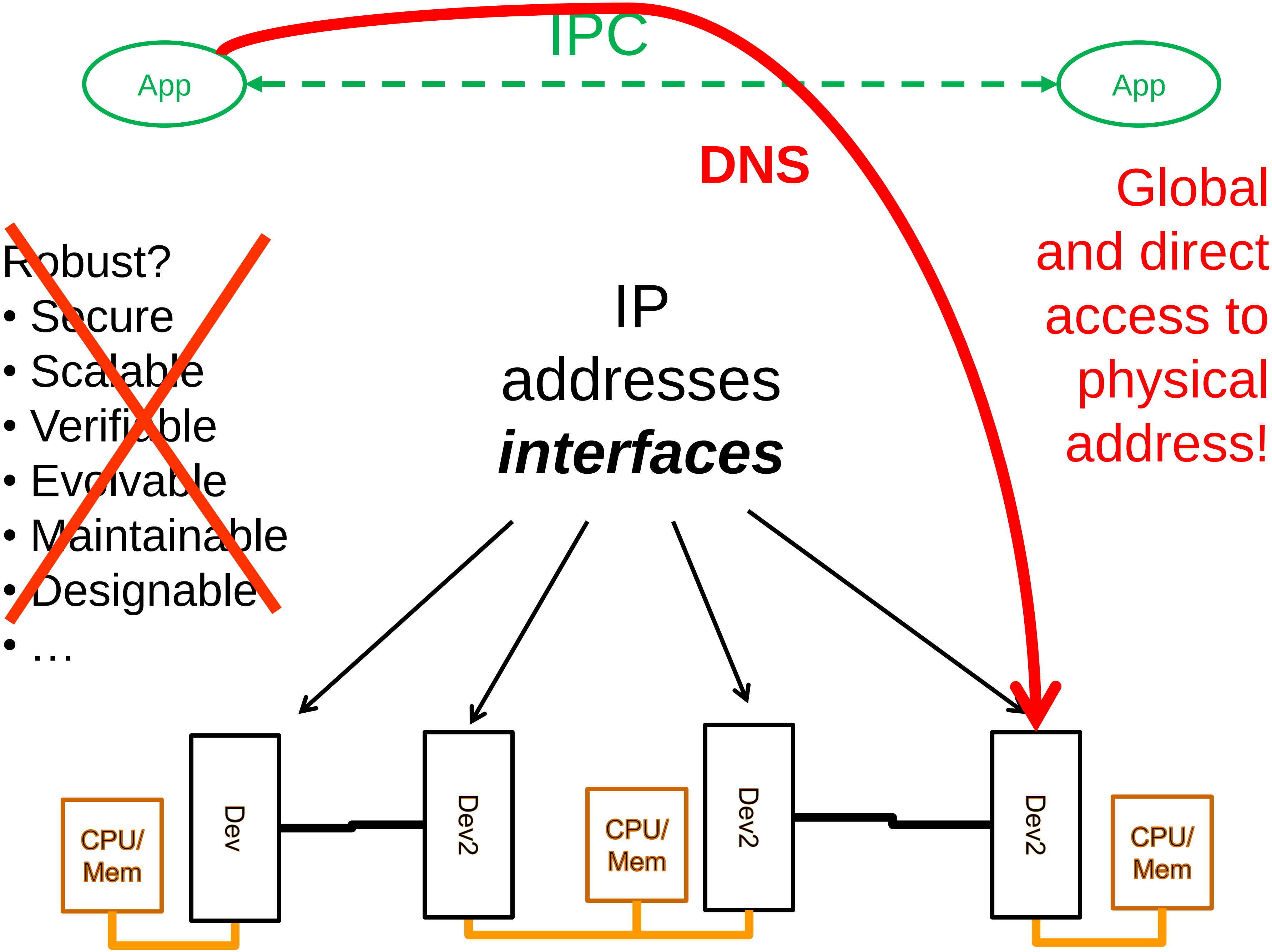
quantization
for robustness



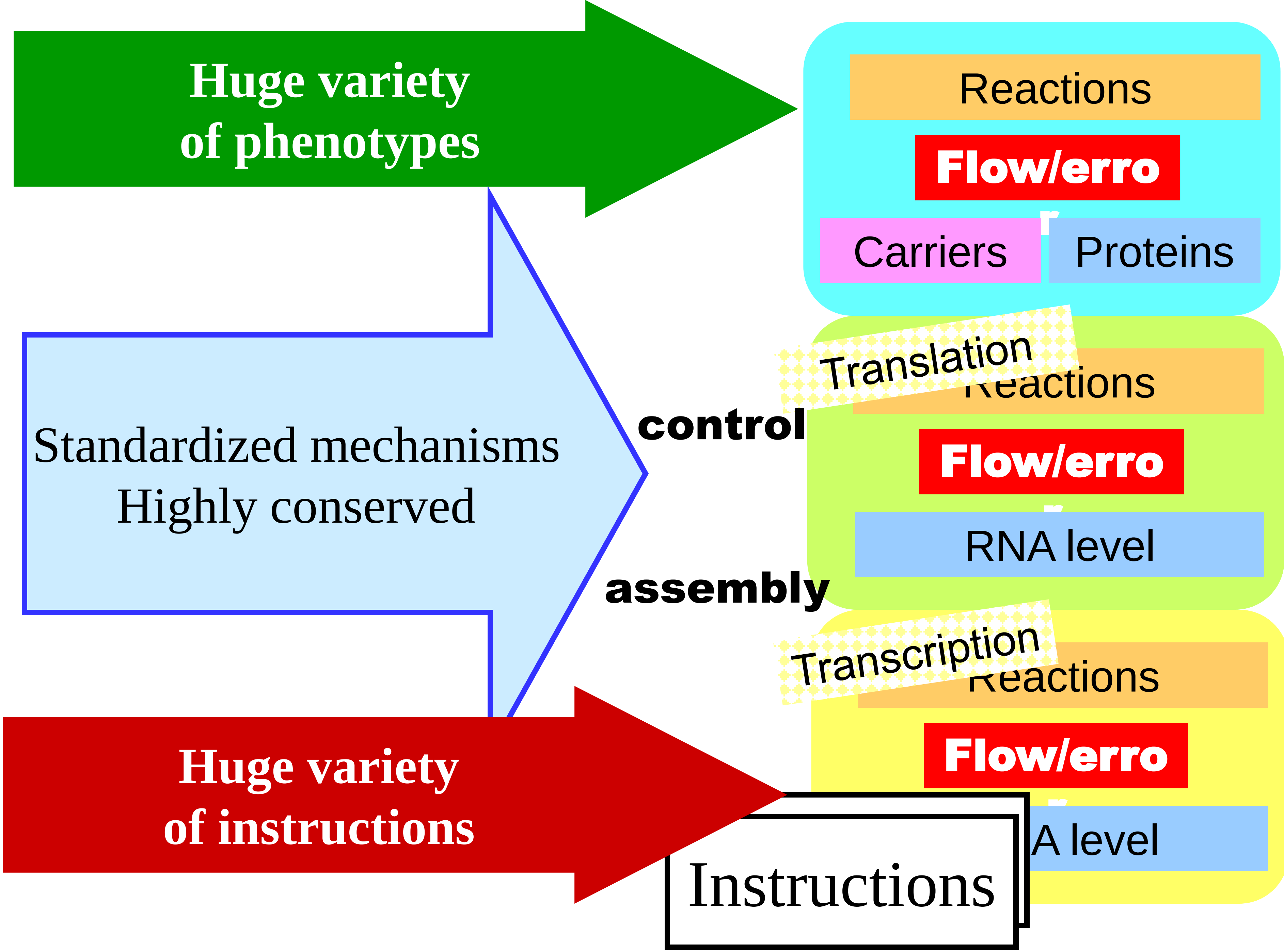
Cells have limited
and structured
access to DNA
layer

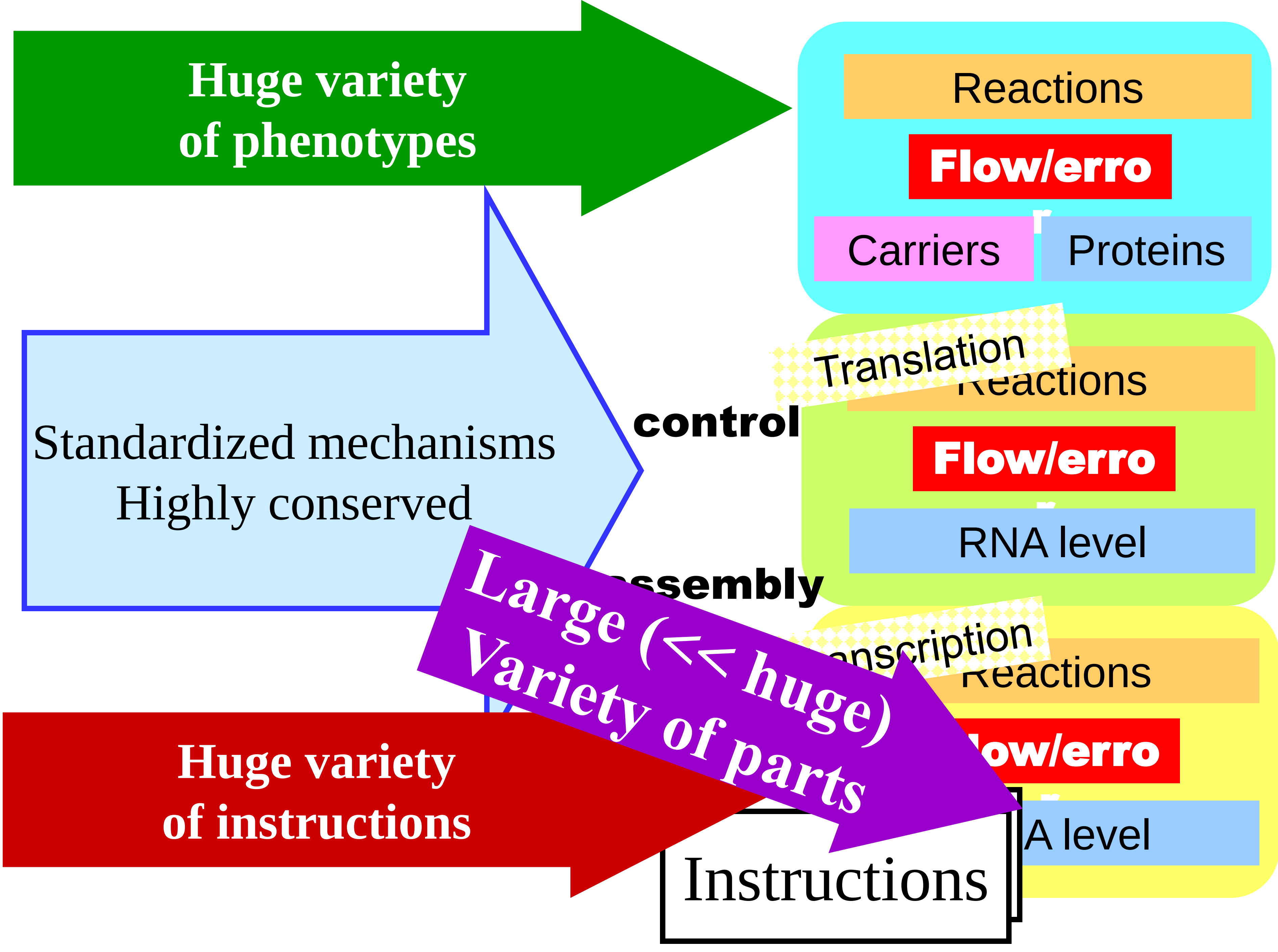
constraints on
cross-layer
interactions

Prevents unraveling of lower layers



IPC
App
App
DNS
Global and direct access to physical address!
Robust?
• Secure
• Scalable
• Verifiable
• Evolvable
• Maintainable
• Designable
• …
IP addresses *interfaces*
Dev
Dev2
Dev2
Dev2
CPU/ Mem
CPU/ Mem
CPU/ Mem

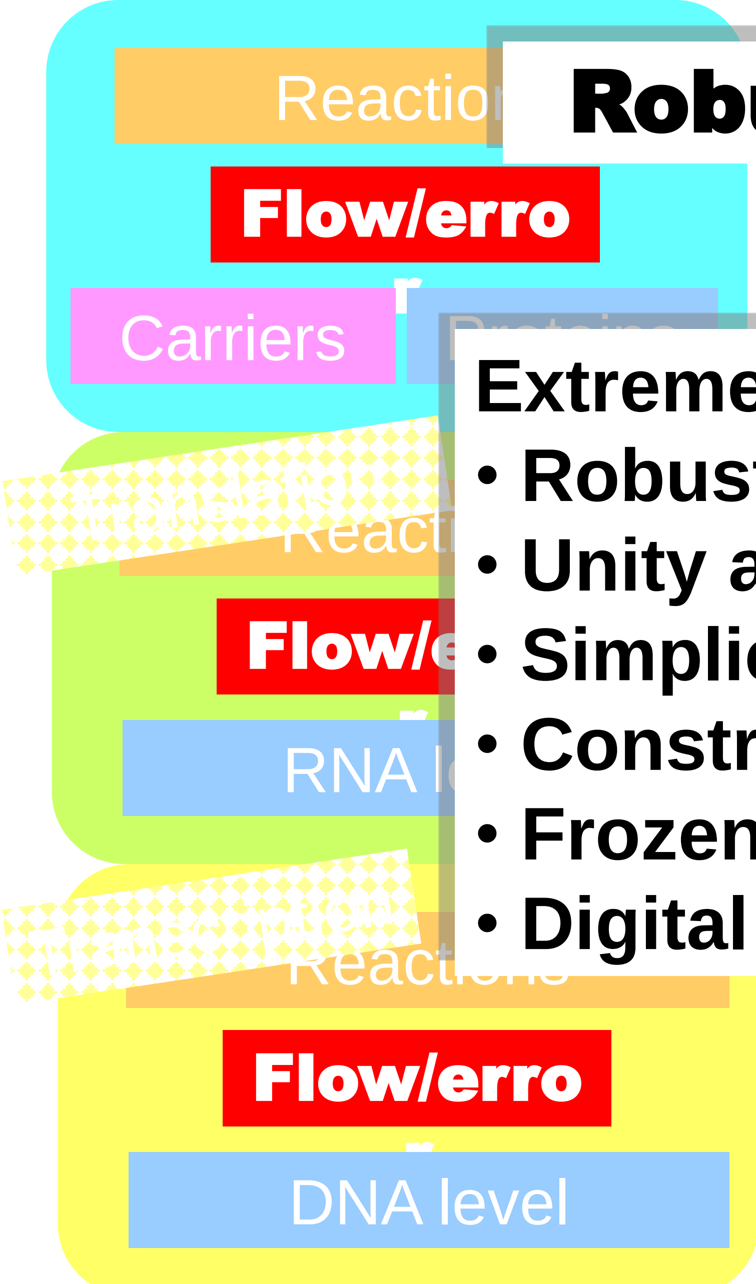

Huge variety
of phenotypes
Reactions
Flow/erro
r
Carriers
Proteins
Translation
Reactions
control
Flow/erro
r
RNA level
Standardized mechanisms
Highly conserved
assembly
Transcription
Reactions
Flow/erro
r
A level
Huge variety
of instructions
Instructions

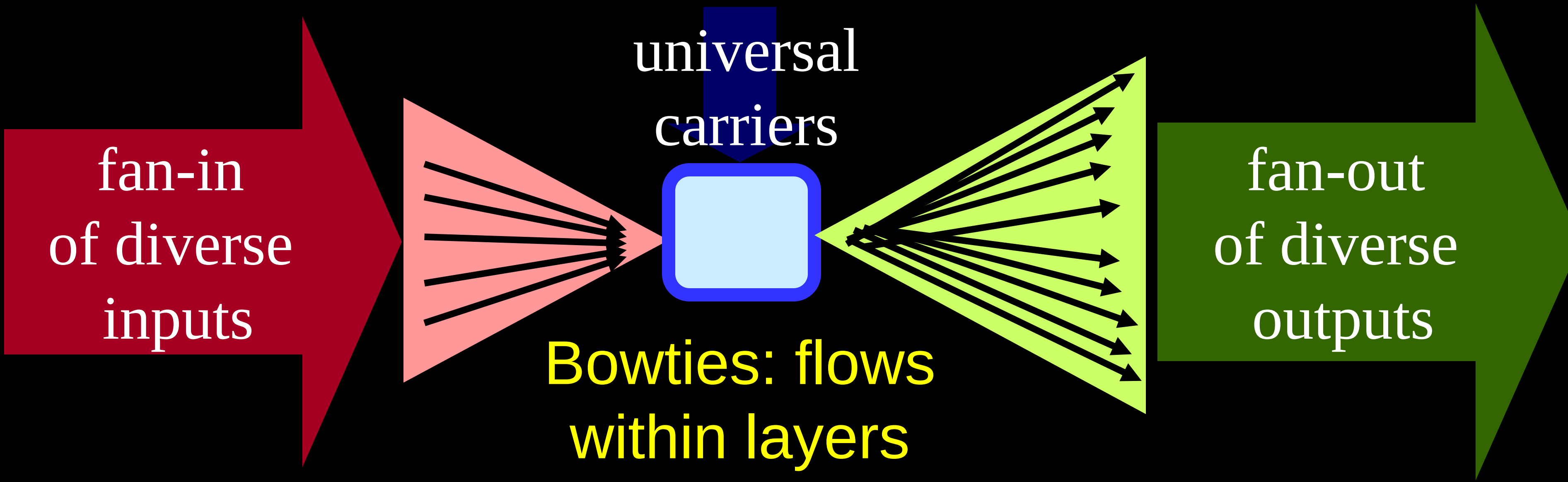

Huge variety
of phenotypes
Reactions
Flow/erro
r
Carriers
Proteins
Translation
Reactions
control
Flow/erro
r
RNA level
Standardized mechanisms
Highly conserved
assembly
anscription
Reactions
Flow/erro
r
A level
Large (<< huge)
Variety of parts
Huge variety
of instructions
Instructions

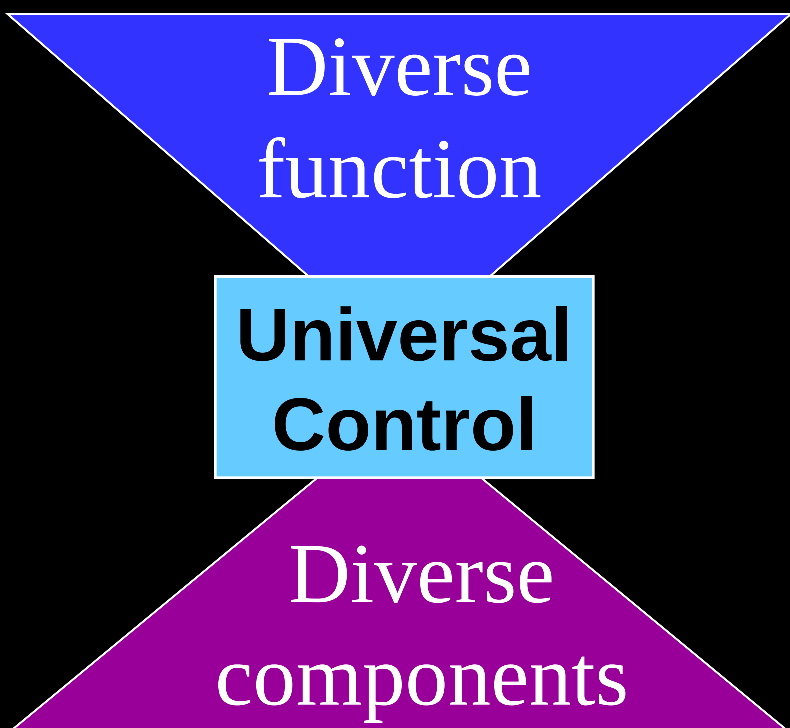
Robust yet fragile

Extremes of

- Robust yet fragile
- Unity and diversity
- Simplicity and complexity
- Constrained and flexible
- Frozen and evolvable
- Digital and analog



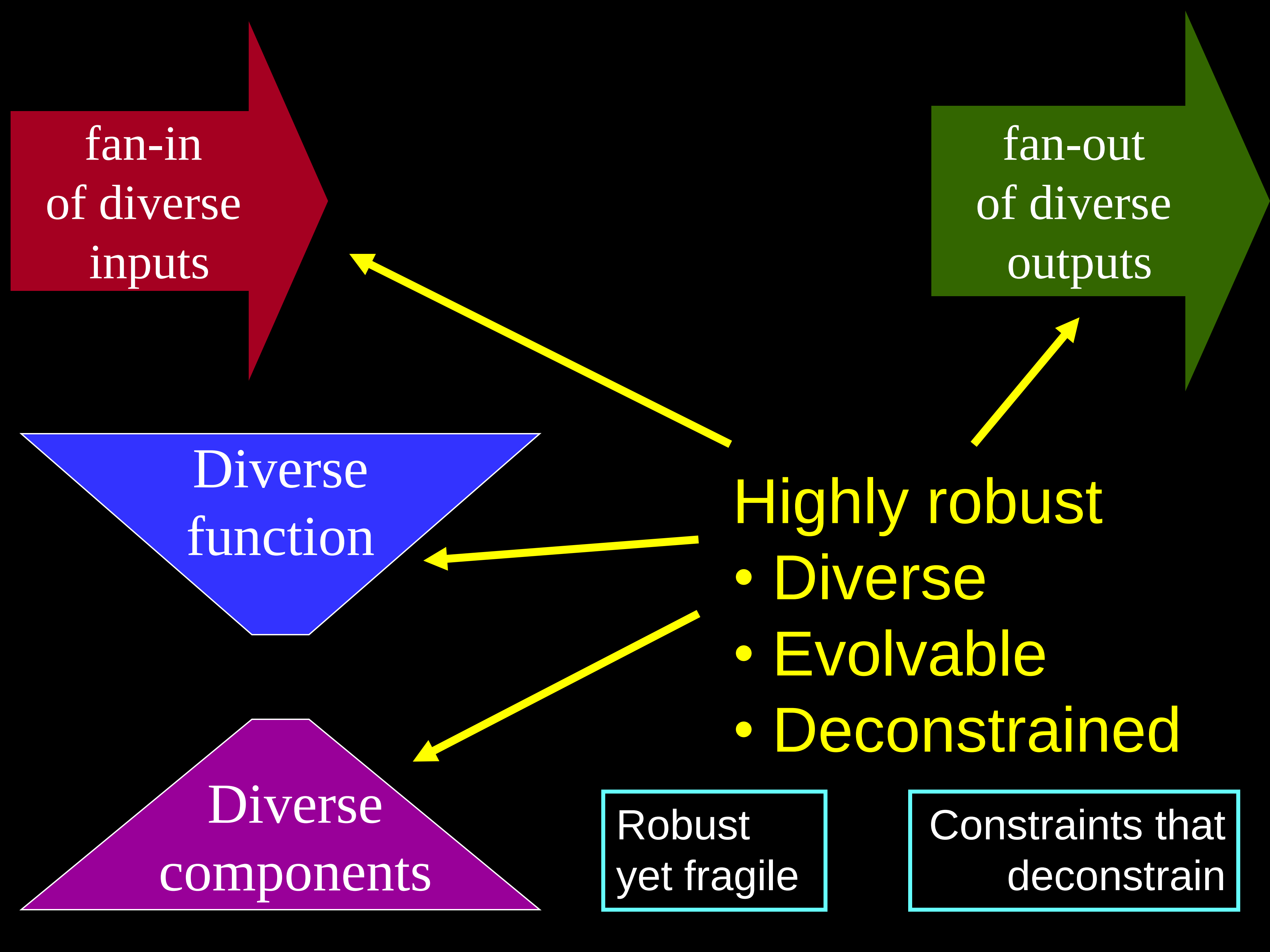




Essential ideas

Robust yet fragile

Constraints that deconstrain


fan-in of diverse inputs
fan-out of diverse outputs
Diverse function
Diverse components
Highly robust
• Diverse
• Evolvable
• Deconstrained
Robust yet fragile
Constraints that deconstrain

universal
carriers

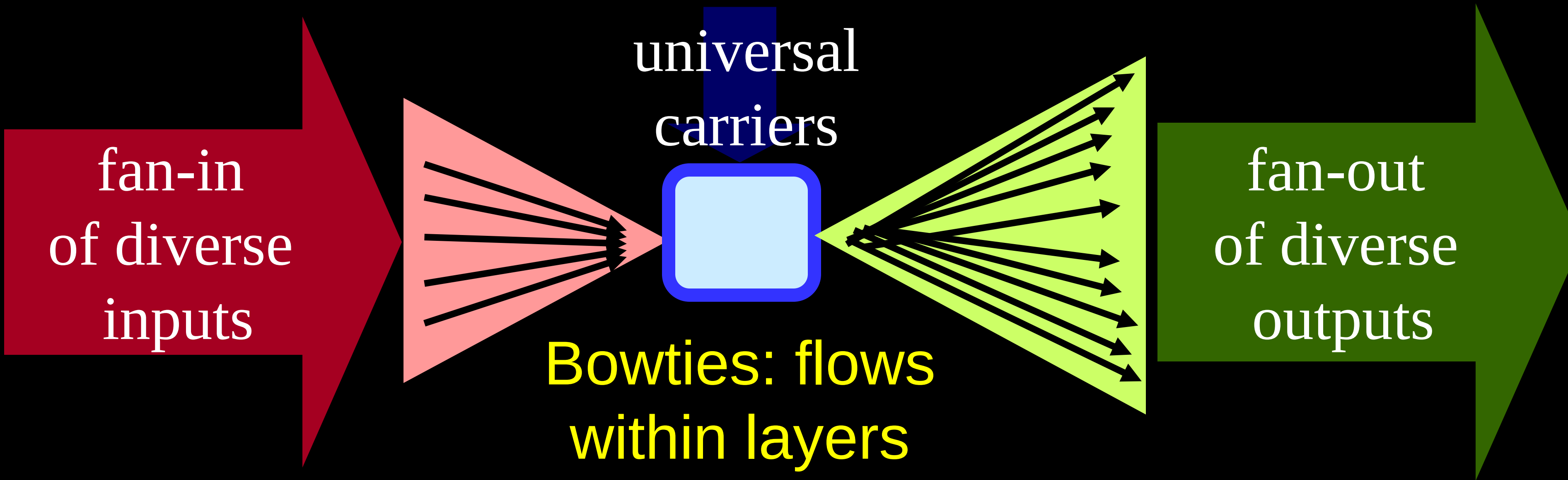
Highly fragile

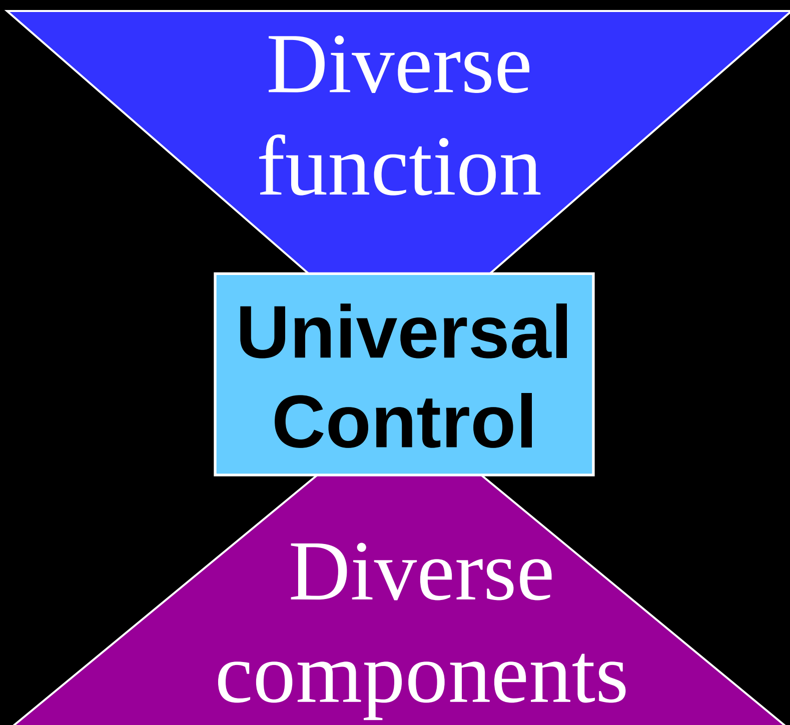
- Universal
- Frozen
- Constrained

Universal Control

Robust
yet fragile

Constraints that
deconstrain





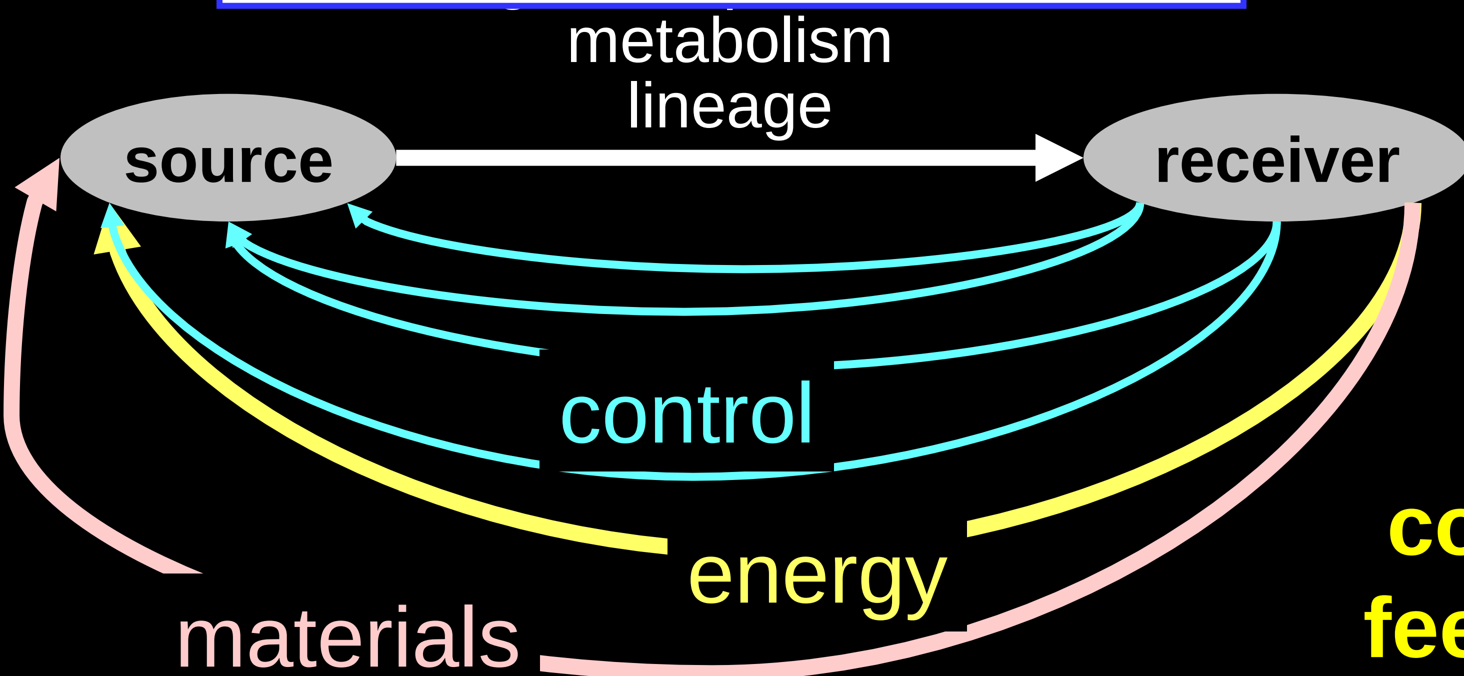
Essential ideas

Robust yet fragile

Constraints that deconstrain

What theory is relevant to these more complex feedback systems?

$$\frac{1}{\pi}\int_0^\infty \ln\left|S(j\omega)\right|\frac{z}{z^2+\omega^2}d\omega \geq \ln\left|\frac{z+p}{z-p}\right|$$



More complex feedback